
Universidade de Aveiro Departamento de Eletrónica,
2015 Telecomunicações e Informática

Rui Marcos
Brandão Antunes

Música Genómica
Genomics Music

Rui Marcos Brandão Antunes

Música Genómica
Genomics Music

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Carlos Alberto da Costa Bastos, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e da Doutora Vera Mónica Almeida Afreixo, Professora Auxiliar do Departamento de Matemática da Universidade de Aveiro

aos meus padrinhos
aos meus avós
aos meus pais

o júri

presidente	**Doutor Rui Manuel Escadas Ramos Martins** Professor auxiliar da Universidade de Aveiro
vogal	**Doutor José Carlos Silva Cardoso** Professor associado da Universidade de Trás-Os-Montes e Alto Douro
orientador	**Doutor Carlos Alberto da Costa Bastos** Professor auxiliar da Universidade de Aveiro

Agradecimentos

São muitas as pessoas que eu gostaria de agradecer por tornar este trabalho possível. Primeiramente aos meus orientadores, Doutor Carlos Alberto da Costa Bastos e Doutora Vera Mónica Almeida Afreixo por toda a sua ajuda, paciência e disponibilidade. Queria expressar a minha gratidão por todas as sugestões dadas pelos meus orientadores, uma vez que muitas ideias deste trabalho surgiram dessas sugestões. Por último, gostaria de agradecer à minha família e aos meus amigos por todo o apoio e incentivo.

Palavras-chave Bioinformática, ADN, Música, Distância entre palavras, Síntese e avaliação musical

Resumo

Dada a grande informação genética já descoberta ao longo do tempo, a música genómica permite uma análise um pouco invulgar, que consiste na conversão de sequências de ADN em música. Na expectativa que será possível detetar através da audição, caraterísticas interessantes no ADN. Vários trabalhos já foram realizados nesta área.

O objetivo principal deste trabalho consiste no desenvolvimento de algoritmos adequados, com os quais é possível criar músicas a partir de dados simbólicos, dados estes que são obtidos através de sequências de nucleótidos de vários organismos. Neste trabalho os algoritmos de conversão usam mapeamentos relacionados com as distâncias entre palavras e as frequências de ocorrência das palavras.

A síntese de áudio digital é também abordada neste trabalho. Avaliou-se de forma objetiva e subjetiva as músicas convertidas a partir do ADN.

O resultado final é uma aplicação, de simples utilização para o utilizador, com vários parâmetros de entrada ajustáveis nos algoritmos de conversão, permitindo uma grande liberdade na criação da música através do ADN.

Keywords Bioinformatics, DNA, Music, Distance between words, Synthesis and musical evaluation

Abstract Given the considerable genetic information already discovered over time, genomic music allows a somewhat unusual analysis, which consists in the conversion of DNA sequences in music. In the expectation that it will be possible to detect by listening, interesting features in DNA. Several studies have been conducted in this area.
The main objective of this work is to develop suitable algorithms with which you can create music from symbolic data, these data are obtained from nucleotide sequences of various organisms. In this work the conversion algorithms use mappings related to the distances between words and the frequency of occurrence of words.
Digital audio synthesis is also discussed in this work. Songs converted from DNA were evaluated objectively and subjectively.
The final result is an application of simple use, for the user, with various adjustable input parameters of the conversion algorithms, allowing a great freedom in the creation of music through the DNA.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

A	Adenina
ADN	Ácido DesoxirriboNucleico
C	Citosina
EBI	*The European Bioinformatics Institute*
ECG	*Equivalent Composition Group*
EMBL	*European Molecular Biology Laboratory*
ENA	*European Nucleotide Archive*
G	Guanina
MATLAB	*Matrix Laboratory (Mathworks, Inc.)*
MIDI	*Musical Instrument Digital Interface*
MMA	*Midi Manufacturers Association*
NCBI	*National Center for Biotechnology Information*
T	Timina

Capítulo 1

Introdução

O ácido desoxirribonucleico (ADN) contém a informação genética de cada ser vivo, que é transmitida hereditariamente. Apesar de várias descobertas já terem sido feitas, ainda há muito a descobrir sobre o mesmo.

Já foram feitos vários estudos sobre a transformação do ADN em música, *Susumu Ohno* [Ohno and Ohno, 1986,Ohno, 1987,Ohno, 1993] foi o pioneiro nesta área e verificou o princípio de repetição tanto no ADN como na música, ou seja, tanto um como o outro estão cheios de periodicidades (redundância).

Já noutro trabalho [Sánchez Sousa et al., 2005] são apresentadas algumas conversões entre sequências de ADN e sequências musicais: na conversão mais simples é apenas usada uma escala de quatro notas musicais que correspondem aos quatro nucleótidos (A, C, G e T), sem tomar em consideração outras caraterísticas. Outra técnica mais avançada também desenvolvida no mesmo trabalho é o uso de vários genes produzindo uma única música, correspondendo cada gene a uma pauta musical, assim incorporando todos os genes tem-se uma partitura com várias pautas musicais. Neste trabalho o ritmo e o estilo da melodia não foram codificados a partir do gene, mas foram impostos por fatores externos dando assim mais "vida" à musica.

Outro trabalho [Takahashi and Miller, 2007] tem como objetivo a conversão de sequências codificantes de um genoma em notas musicais para revelar padrões auditivos, incorporando também ritmo nas notas. Inicialmente usaram um intervalo de 20 notas musicais, onde cada aminoácido correspondia a uma nota musical, depois melhoraram a musicalidade atribuindo um acorde musical a cada aminoácido, em vez de uma única nota musical. Para definir o ritmo atribuíram uma de quatro durações e usaram a frequência do codão (por 1000 ocorrências) para ditar a sua duração.

No trabalho [Gena and Strom, 1995] são contempladas conversões de genomas completos de vírus, genomas parciais de bactérias e sequências completas de proteínas humanas. Neste trabalho foram usadas extensivamente as propriedades físicas e químicas dos aminoácidos (constante de dissociação, peso molecular e classe química) e as propriedades dos nucleótidos (temperaturas de fusão) para a criação de música. As notas musicais, durações e intensidades são determinadas através de funções pré-definidas.

O trabalho [Ingallsa et al., 2009] é baseado nos alinhamentos de sequências de ADN, pois são uma técnica muito importante na área da genética, desenvolveram uma ferramenta para ouvir dados genómicos em grande escala. Permite ouvir uma representação musical do alinhamento de um genoma.

Neste capítulo segue-se a motivação para a realização deste trabalho, uma ligeira introdução a alguns conceitos biológicos e musicais, os objetivos gerais deste trabalho e a estrutura da dissertação.

1.1 Motivação

Desde há muito tem-se estudado o ADN, e se tenta compreender a sequenciação dos nucleótidos no genoma. Usando uma área diferente, a música, espera-se que a análise musical permita explorar caraterísticas interessantes do ADN. Na expectativa que o nosso sentido auditivo e capacidade musical consiga alertar-nos sobre determinadas caraterísticas relevantes para a genética. Assim é possível ouvir a informação genética.

Apesar das sequências de ADN serem habitualmente representadas pelos símbolos A, C, G e T, nem sempre a análise dos dados é realizada diretamente sobre estes símbolos, normalmente são realizados um ou mais mapeamentos de forma a obter outros dados equivalentes, nesta dissertação são usadas extensivamente as **frequências de ocorrência** das palavras[1] em determinadas janelas[2] e um mapeamento que surge dos trabalhos [Nair and Mahalakshmi, 2005, Afreixo et al., 2009], denominado de **distância entre palavras**.

Depois de realizados variados mapeamentos e transformações dos símbolos em dados numéricos, a informação numérica é convertida através de um algoritmo (com parâmetros de entrada adequados) em música, são determinadas através de funções desenvolvidas neste trabalho as suas notas musicais, durações e intensidades.

Surge então a necessidade da criação de algoritmos computacionais eficientes que permitam ao utilizador a escolha de determinados parâmetros de entrada (entre eles qual a sequência a converter, possibilidade de seleção de uma determinada região código ou não-código, etc.) para a criação da música, tendo como objetivo a eficácia e facilidade da realização de "experiências musicais".

[1] Considera-se por palavra um símbolo ou um índice, ou um determinado conjunto de símbolos ou índices que dizem respeito a um oligonucleótido, um aminoácido ou a um conjunto de palavras do mesmo grupo *ECG* (mais informação na secção 3.1 do capítulo 3).

[2] Janela engloba um determinado número de nucleótidos, e pode-se dizer que o tamanho da janela é de N-oligonucleótidos (por exemplo, se a janela conter 12 nucleótidos, então é o mesmo dizer que o tamanho da janela é de 6 di-nucleótidos, 4 tri-nucleótidos ou 3 tetra-nucleótidos).

1.2 Conceitos biológicos

O ácido desoxirribonucleico (ADN) é um composto orgânico cujas moléculas contém a informação genética. A estrutura do ADN foi descoberta em 1953 por James Watson e por Francis Crick [Watson and Crick, 1953]. Toda a a informação hereditária de um organismo (codificada no seu ADN) denomina-se de genoma. O genoma inclui os genes e as sequências não-codificantes.

O ADN pode ser visto como uma sequência de nucleótidos, cuja sequência constitui a informação genética. Os nucleótidos são quatro:

- adenina (A);
- citosina (C);
- guanina (G);
- timina (T).

Em genética, um oligonucleótido é um conjunto de N-nucleótidos, por exemplo di-nucleótidos ($N = 2$), tri-nucleótidos ($N = 3$) ou tetra-nucleótidos ($N = 4$). Um codão é um conjunto de três nucleótidos que codifica um determinado aminoácido, podendo indicar o ponto de início ou fim de uma sequência. Um codão pode tomar uma de 64 combinações possíveis ($4^3 = 64$), no entanto só existem 20 aminoácidos (21 considerando o marcador de fim da sequência - *stop*), esta diferença existe porque a cada aminoácido pode corresponder um ou mais codões, ou seja, não é uma transformação biunívoca.

1.3 Conceitos musicais

A música é representada através de uma partitura que pode ter uma ou mais pautas, cada pauta é representada por 5 linhas horizontais, é preciso também apresentar a clave, o compasso, as notas e eventualmente outros pormenores musicais relacionados com articulação e intensidade musical. O som relativo às notas musicais é constituído por ondas sonoras com as seguintes caraterísticas:

- altura: consiste na frequência da onda sonora, indica a nota musical;
- duração: dada pelo tempo de duração da onda sonora, indica a figura musical;
- intensidade: dada pela amplitude da onda sonora, indica a dinâmica musical;
- timbre: consiste no formato da forma de onda (que está relacionado com todas as componentes de frequências), é o que distingue os diversos instrumentos musicais, a voz humana, etc. O envelope sonoro está relacionado com a perceção do timbre, consiste na variação da amplitude da onda sonora ao longo do tempo: é lógico pensar que a amplitude da onda sonora decresce ao longo do tempo devido à propagação no meio impor uma atenuação mais ou menos acentuada.

Existem 12 notas musicais (ver figura 1.1), são elas:

Dó - Dó♯/Ré♭ - Ré - Ré♯/Mi♭ - Mi - Fá - Fá♯/Sol♭ - Sol - Sol♯/Lá♭ - Lá - Lá♯/Si♭ - Si

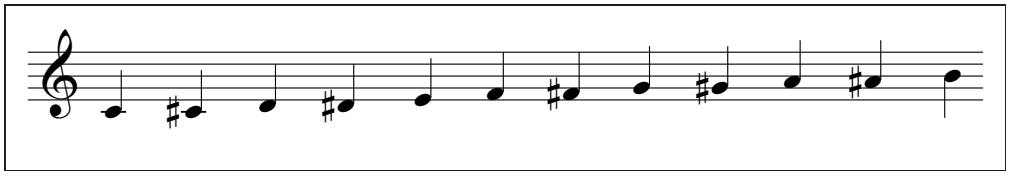

Figura 1.1: Representação das notas musicais.

De notar que nalguns casos a mesma nota tem dois nomes diferentes, é o caso de Dó♯ e Ré♭ por exemplo. Na figura 1.1 apenas foram apresentadas as notas centrais, no entanto também existem as mesmas notas com frequências superiores e inferiores. Por exemplo, diz-se que a nota X está uma oitava abaixo da nota Y se a frequência da onda X for metade da frequência da onda Y, neste caso as notas são as mesmas, apenas a sua altura é alterada (se tiver uma oitava abaixo é um som mais grave, se tiver uma oitava acima é um som mais agudo).

O tom é a unidade de medida para se medir a distância entre notas. O semitom, metade de um tom, é a menor distância possível entre duas notas. Assim, o aumento numa nota de um semitom implica colocar um sustenido (♯) à frente da nota, assim como quando se diminui um semitom numa nota coloca-se um bemol (♭) à frente da nota.

O andamento impõe a "velocidade" da música. O compasso é responsável pela divisão em partes de igual duração na música sendo colocado um traço vertical na partitura entre compassos, é o responsável por impor o número de unidades de tempo por cada parte de igual duração. É o que determina a estrutura rítmica da música. Na fórmula de compasso, o denominador informa o número de partes em que se divide a semibreve, cada parte é denominada a unidade de tempo, e o numerador indica quantas unidades de tempo estão presentes no compasso. Por exemplo no caso do compasso ser $\frac{2}{4}$ a unidade de tempo é a semínima, a semibreve é dividida em quatro partes de durações iguais. Neste exemplo uma semibreve tinha a duração de dois compassos, que é equivalente à duração de quatro semínimas.

As figuras musicais representam a duração das notas musicais, em que as mais comuns são (ver tabela 1.1 e figura 1.2):

Nomes das figuras musicais	**Duração relativa**
Semibreve	1
Mínima	1/2
Semínima	1/4
Colcheia	1/8
Semicolcheia	1/16
Fusa	1/32
Semifusa	1/64

Tabela 1.1: Nomes das figuras musicais mais comuns.

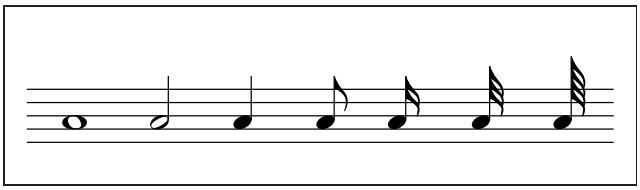

Figura 1.2: Representação das figuras musicais mais comuns.

Os pontos de aumentação permitem ter mais variabilidade de durações, assim é possível colocar um ponto (ou mais) imediatamente a seguir à nota, cada ponto adiciona metade do valor da figura que o precede (por exemplo, adicionando dois pontos de aumentação à frente de uma nota, então a duração será 1.75 vezes superior).

As pausas são o que representa o silêncio, têm nomes iguais aos das figuras musicais, do mesmo modo há símbolos para identificar pausas de durações idênticas às figuras musicais (ver figura 1.3).

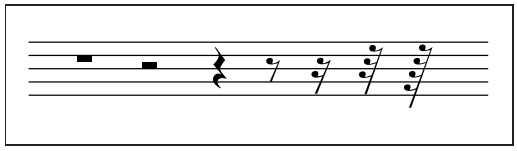

Figura 1.3: Representação das pausas musicais mais comuns.

A intensidade das notas ao longo da pauta musical é indicada sob a forma de siglas, as mais frequentes são:

- ***ppp*** - *pianississimo* - a intensidade é muito, muito baixa;
- ***pp*** - *pianissimo* - a intensidade é muito baixa;
- ***p*** - *piano* - a intensidade é baixa;
- ***mp*** - *mezzo piano* - a intensidade é pouco forte;
- ***mf*** - *mezzo forte* - a intensidade é moderadamente forte;
- ***f*** - *forte* - a intensidade é forte;
- ***ff*** - *fortissimo* - a intensidade é muito forte;
- ***fff*** - *fortississimo* - a intensidade é muito, muito forte.

Existem também outros símbolos para a intensidade musical, indicando crescimento ou decrescimento gradual, ou como um aumento súbito de intensidade numa única nota. Há ainda outras caraterísticas que não serão apresentadas, visto não serem tão relevantes para este trabalho, como por exemplo a possibilidade de durações diferentes ou a articulação musical.

1.4 Objetivos gerais

O trabalho descrito nesta dissertação consiste na conversão de ADN (sequências simbólicas) em música, tendo em mente a deteção de padrões, podendo por exemplo conseguir-se diferenciar regiões codificantes e regiões não-codificantes através da música produzida.

Assim, os objetivos principais desta dissertação são:

- desenvolvimento de algoritmos eficientes para a conversão de sequências de ADN em música;
- criação e ajustamento de mapeamentos coerentes entre dados simbólicos e numéricos, e entre dados numéricos e musicais;
- criação de uma ferramenta desenvolvida em MATLAB, com uma interface de fácil uso, que possibilite a criação de música (a partir de ADN) de forma simples, permitindo ao utilizador variar parâmetros de entrada;
- realização de várias experiências musicais e respetivas conclusões acerca de possíveis semelhanças entre a música e o ADN;
- avaliação musical de forma objetiva e subjetiva.

1.5 Estrutura da dissertação

Como já referido anteriormente, este trabalho consiste principalmente no desenvolvimento de uma aplicação capaz de criar música através de sequências de ADN. Para a criação da música é necessária a sintetização do som (que é apresentada no capítulo 2), para que a música possa ficar gravada num ficheiro reproduzível.

Na conversão do ADN em música, primeiramente são obtidos valores numéricos relacionados com o ADN (estes mapeamentos são apresentados no capítulo 3), só depois é gerada a informação musical através dos valores numéricos (esta conversão é apresentada no capítulo 4). Tendo a informação musical pode-se criar o ficheiro que armazena a música (ver capítulo 2).

Todas as conversões e gravações de informação são auxiliadas pelo uso de uma ferramenta desenvolvida neste trabalho, que tem uma interface de simples utilização (esta ferramenta é apresentada no capítulo 5).

Com o auxílio de todos os programas desenvolvidos e com a realização de um inquérito, foram obtidos resultados que são discutidos no capítulo 6. As conclusões e o trabalho futuro são apresentados no final desta dissertação no capítulo 7.

Esta dissertação está dividida em sete capítulos e um apêndice:

- no presente capítulo (1) é introduzido o tema da dissertação assim como a motivação, são introduzidos alguns conceitos biológicos e musicais, e são apresentados os objetivos gerais;
- o capítulo 2 ("Síntese e avaliação musical") apresenta um pouco de síntese de áudio e de avaliação musical (objetiva). A avaliação musical não é usada apenas com o objetivo de avaliar a música, mas serve também para a criação da música, aperfeiçoando a sua avaliação;
- o capítulo 3 ("Do ADN aos números") apresenta a forma como o ADN é convertido em informação numérica. São explicados os mapeamentos realizados a partir de sequências simbólicas para sequências numéricas, indicando os parâmetros de entrada relacionados com essas conversões;
- o capítulo 4 ("Dos números à música") explica como é realizada a conversão de sequências numéricas em música, e que regras foram criadas na sua conversão para que se obtenha um resultado coerente;
- o capítulo 5 ("Ferramenta: as suas aplicações e a sua interface gráfica") apresenta a aplicação desenvolvida neste trabalho, explica como a informação é gerada e gravada. É apresentado um manual do utilizador que explica de forma simples como utilizar a interface;
- o capítulo 6 ("Resultados") apresenta vários resultados relacionados com os algoritmos de conversão e músicas obtidas através do ADN, como os tempos de processamento, distribuição de probabilidades de determinadas variáveis, e respostas a um inquérito sobre algumas músicas de diferentes genomas;
- o capítulo 7 ("Conclusões e trabalho futuro") indica as principais conclusões obtidas através deste trabalho, e também novas ideias para trabalho futuro como evolução e melhoramento;
- o apêndice A ("Código MATLAB") indica e explica, de uma forma muito minimalista, cada função desenvolvida no âmbito deste trabalho;
- o apêndice B ("Inquérito") apresenta todas as questões que foram colocadas aos inquiridos sobre as caraterísticas das músicas resultantes.

Capítulo 2

Síntese e avaliação musical

É necessário guardar a informação musical obtida através de sequências de ADN, para que esta possa ser convertida em música reproduzível. Para criar a música é necessário sintetizar som. Portanto, um dos objetivos deste capítulo é apresentar os conceitos básicos de síntese de áudio, ou seja, a criação do som digital correspondente a instrumentos musicais. Foram estudados os artigos [Petersen, 2004,Petersen, 2001] e os livros [Henrique, 2002,Russ, 2004,Miranda, 2002]. Algumas das caraterísticas desejadas da aplicação desenvolvida neste trabalho eram a sua facilidade de uso e a sua rapidez de execução. Desse modo, com o objetivo de tornar a ferramenta mais eficiente, a aplicação não é responsável pela sintetização do som, mas guarda apenas a informação musical já num formato padronizado (*General MIDI* [MMA, 2015]). Apenas no momento de reproduzir esse ficheiro que contém a informação musical, é que o som é sintetizado pelo dispositivo que está a tentar reproduzir o ficheiro.
De forma a conseguir caraterizar diferentes músicas que foram convertidas de diferentes sequências simbólicas, estudou-se as distribuições de probabilidades de algumas caraterísticas musicais. Para isso é abordada a lei de Zipf [Zipf, 1949, Zipf, 1935] como avaliador musical [Lo, 2012].

2.1 Síntese musical

O som é uma onda mecânica, uma perturbação que se propaga num meio elástico que origina flutuações de pressão, dando origem a um movimento ondulatório. Esta onda pode ser representada através de um sinal elétrico que pode ser armazenado em formato digital. Como não é possível armazenar informação infinita, é necessário fazer uma amostragem do sinal, ou seja converte-se o sinal (função contínua no tempo) para uma função discreta no tempo. A amostragem consiste em recolher amostras em determinados instantes de tempo, normalmente igualmente espaçados. A amplitude de um sinal varia entre valores infinitos, portanto é necessário realizar uma quantização ajustada de acordo com a qualidade sonora desejada. Após a amostragem e a quantização do sinal, este pode ser guardado em formato digital.
Um sinal pode ser representado no domínio do tempo ou da frequência. Dado um sinal representado no domínio do tempo, pode-se obter o sinal no domínio da frequência através da transformada de Fourier. O sinal resultante no domínio da frequência tem componentes reais e imaginárias, muitas vezes é só representado o seu módulo, guardando apenas a informação das amplitudes das frequências e perdendo a informação sobre as fases das frequências. Na figura 2.1 é representada uma sinusoide no domínio do tempo e da frequência, no domínio da

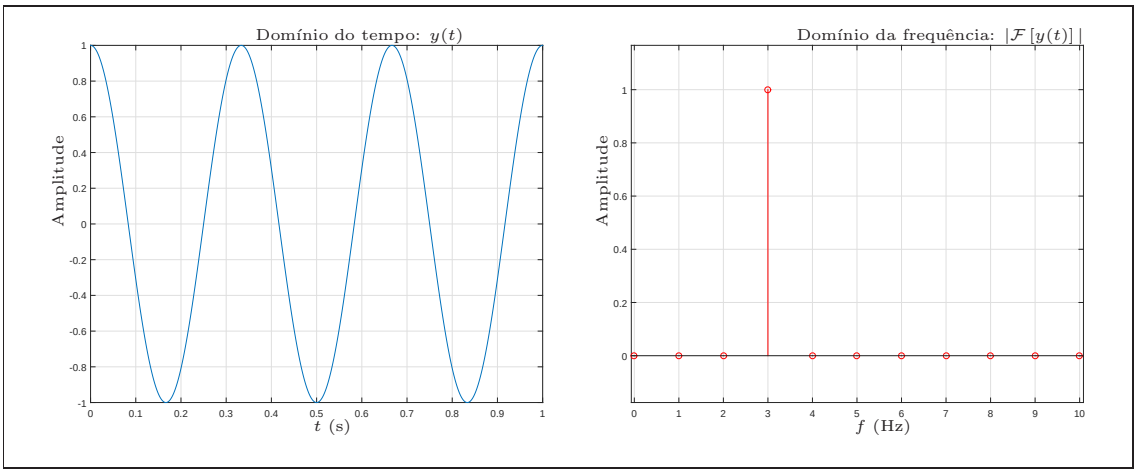

Figura 2.1: Representação da função $y(t) = \cos(wt)$, com $w = 2\pi f$ e $f = 3$ Hz no domínio do tempo (figura da esquerda) e da frequência (figura da direita).

---

frequência apenas é apresentado o módulo da transformada de Fourier. A representação do módulo no domínio da frequência permite apenas conhecer as frequências presentes num determinado sinal e as respetivas amplitudes. Uma representação num plano de tempo e frequência permite conhecer as frequências e os seus momentos de ocorrência, mas não permite saber quais as suas amplitudes, para isso é necessária uma representação com três dimensões (tempo, frequência e amplitude). Na figura 2.2, estão representadas duas sinusoides consecutivas de frequências diferentes no domínio do tempo e num plano de tempo e frequência. A representação no plano de tempo e frequência tem a desvantagem de não incluir as amplitudes do sinal, neste caso as amplitudes são desconhecidas.

Para a criação do som de instrumentos musicais são usados sinais sinusoidais. Um sinal sinusoidal é geralmente representado pela função:

$$x(t) = A\cos(wt + \phi)$$

em que A é a amplitude do sinal, w é a frequência angular (em radianos por segundo), ϕ é a fase inicial (em radianos) e t é a variável independente que representa o tempo. A frequência angular w é igual a $2\pi f$, em que f representa a frequência em Hertz.

Um determinado som de um instrumento é constituído por várias frequências com diferentes amplitudes, ou seja pode ser considerado como um somatório de sinusoides (ver por exemplo figura 2.3, onde D é o resultado do somatório de três sinusoides A, B e C).

Para definir um sinal $x(t)$ no domínio da frequência é necessário recorrer à transformada de Fourier, esta operação consiste em decompor uma função representada no domínio do tempo numa soma de componentes com várias frequências. Assim, a transformada de Fourier permite descobrir quais as frequências de um determinado sinal. A transformada de Fourier é definida pela seguinte expressão:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-2\pi \mathrm{i} ft}\mathrm{d}t \equiv \mathcal{F}[x(t)]$$

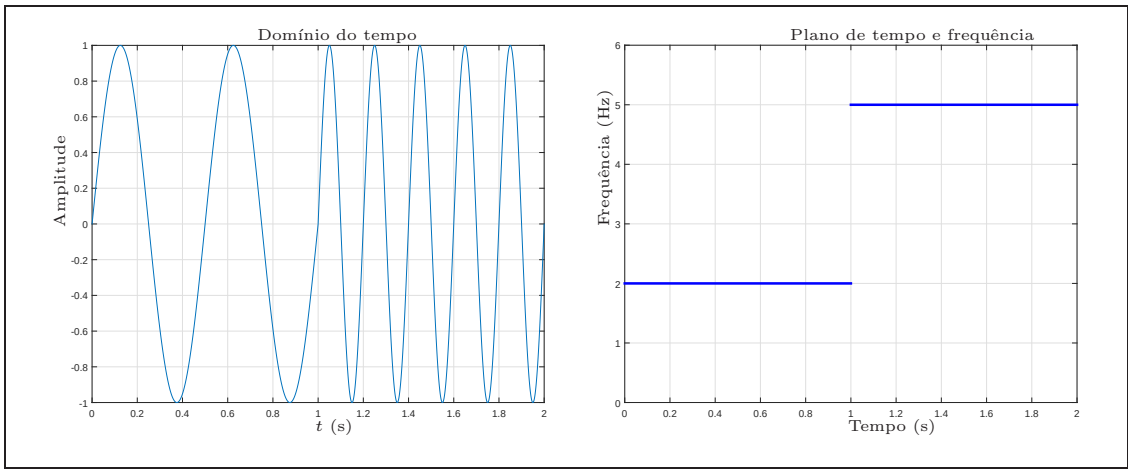

Figura 2.2: Representação de duas sinusoides de 2 e 5 Hz respetivamente, num plano de tempo e frequência: inicialmente existe uma sinusoide com frequência de 2 Hz durante o primeiro segundo, e no segundo a seguir existe apenas uma sinusoide de 5 Hz. Notar que no plano de tempo e frequência as amplitudes das frequências são desconhecidas.

Um som pode ser analisado observando o seu espetro de frequência para se concluir sobre a distribuição das suas componentes de frequências. O conhecimento das amplitudes das componentes de frequências de uma determinada nota musical num dado instrumento musical é a base para a construção da forma de onda sintetizada.

Conhecendo as frequências f_n e as amplitudes a_n respetivas, associadas à n-ésima componente de frequência, então a forma de onda sintetizada pode ser escrita sob a forma:

$$y(t) = \sum_{n=1}^{N} a_n \cos(2\pi f_n t + \phi_n)$$

Esta é uma aproximação bastante simples feita através de síntese aditiva. Notar que uma determinada nota musical de um determinado instrumento contém frequências que não são múltiplas da frequência fundamental.

Cada instrumento produz um som com uma determinada forma de onda e um determinado envelope sonoro (timbre). Um som de um instrumento pode ser dividido em três períodos de duração distintos (como mostra a figura 2.4):

- transitório de ataque;
- período de estabilidade (regime estacionário);
- transitório de extinção (decaimento final).

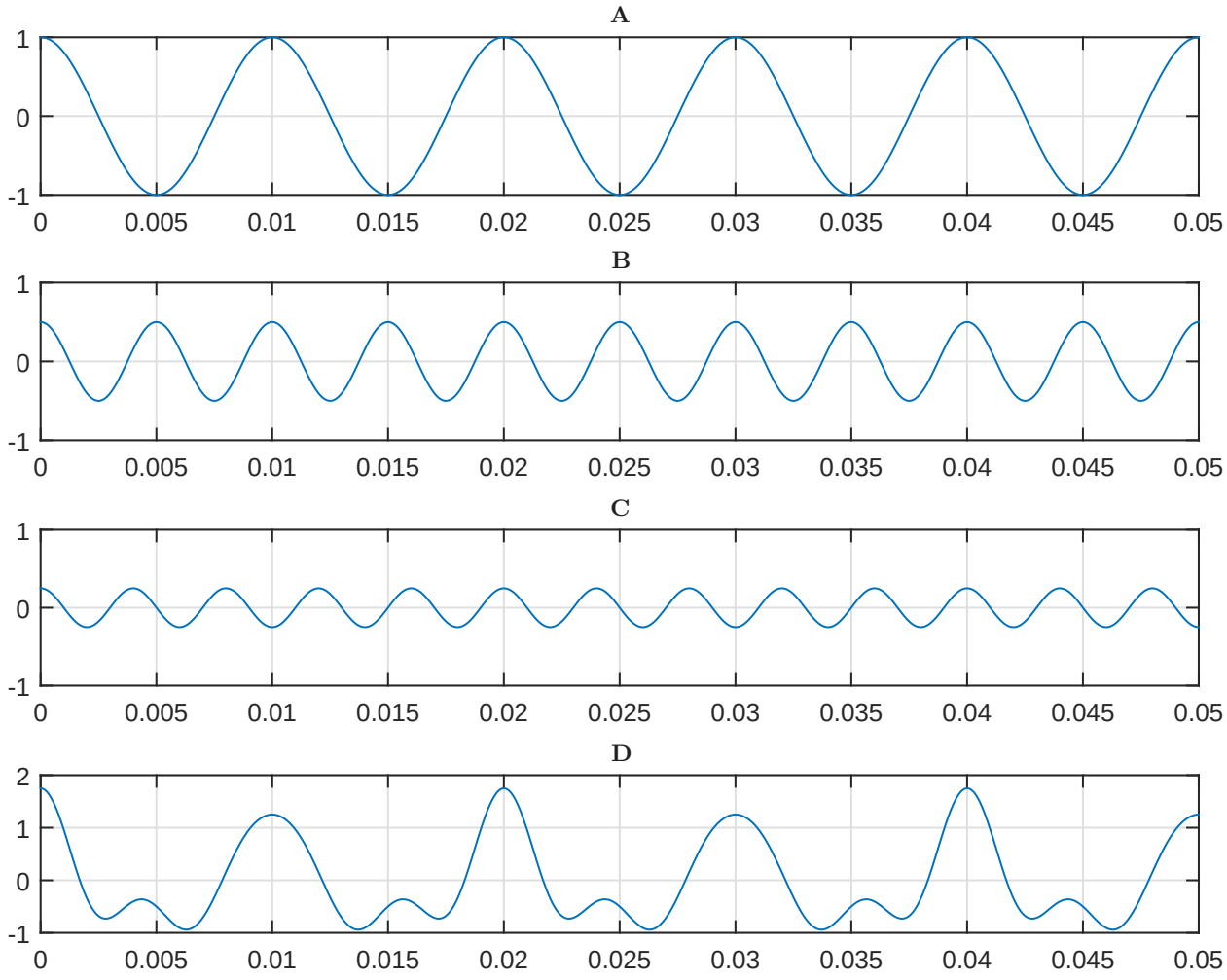

Figura 2.3: Representação de três sinusoides com amplitudes e frequências diferentes e a respetiva soma. **A** - sinusoide de 100 Hz com amplitude 2; **B** - sinusoide de 200 Hz com amplitude 1; **C** - sinusoide de 250 Hz com amplitude 0.5; **D** - somatório das sinusoides anteriores.

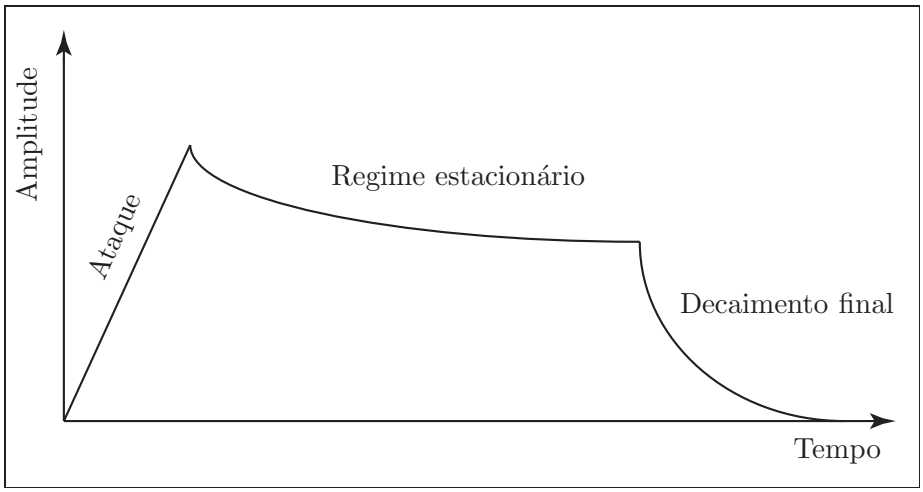

Figura 2.4: Envelope sonoro (figura adaptada do livro [Henrique, 2002]).

A técnica de somar sinusoides por forma a obter um determinado som complexo é designada de síntese aditiva. Esta técnica é bastante flexível, no entanto é exigente do ponto de vista de processamento. O uso do padrão *MIDI* (que é de seguida apresentado) permite gravar informação musical e dispensa a aplicação da função de sintetizar áudio, pelo que torna a aplicação mais rápida e eficiente.

2.2 *General MIDI*

O *General MIDI (Musical Instrument Digital Interface)* [MMA, 2015] é uma codificação padronizada que permite gravar informação musical indicando quais os instrumentos musicais a usar. Os arquivos *MIDI* (extensão .mid) contém dados como as notas, durações, intensidades, etc., ou seja, estes arquivos não contém uma forma de onda, mas sim uma indicação da(s) sequência(s) de notas musicais e respetivas caraterísticas. Com este formato o ficheiro ocupa menos espaço relativamente a um ficheiro que guarda a forma de onda sonora. Para ser possível ouvir um ficheiro *MIDI* é necessário um sintetizador que converta a informação musical em música.

A realização do sintetizador é um compromisso entre a qualidade do som, o tempo de processamento e a memória. Os dispositivos que reproduzem ficheiros *MIDI* podem sintetizar o som por algumas das seguintes técnicas [Russ, 2004, Miranda, 2002, Henrique, 2002]:

- síntese aditiva: consiste na soma de um determinado número de componentes (frequentemente sinusoides);
- técnica por amostras de sons reais: consiste em gravar os sons reais que se pretendem usar;
- síntese por modelação física: o som é gerado através das equações físicas do sistema.

Há 128 notas distintas que podem ser usadas (numeradas de 0 a 127, ver tabela 2.1), a especificação *MIDI* define a nota número 60 como o Dó central, todas as outras são definidas relativamente a essa. Uma música no padrão *GM* (*General MIDI*) é composta por 16 canais (numeros de 0 a 15, em que cada canal corresponde a um instrumento) e tem possibilidade de escolher de um conjunto de 128 instrumentos distintos [MMA, 2015], o canal 10 é reservado exclusivamente apenas para a percussão. A intensidade de cada nota musical também é indicada num valor entre 0 e 127 (inclusive), sendo que 127 corresponde ao maior valor de intensidade. Na especificação de cada nota musical é indicado também o tempo de início e o tempo de fim respetivo, a diferença entre estes tempos corresponde à respetiva duração.

Oitava	Dó	Dó♯	Ré	Ré♯	Mi	Fá	Fá♯	Sol	Sol♯	Lá	Lá♯	Si
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

Tabela 2.1: Número das notas *MIDI* para diferentes oitavas.

Sendo o formato *MIDI* apenas um padrão organizado com informação musical, é então fácil converter informação musical (notas, durações e intensidades) para um ficheiro do tipo .mid (foram usados os programas do trabalho desenvolvido em [Schutte, 2015]). Deixa-se o trabalho de sintetização do som ao recetor do ficheiro *MIDI* (usa por exemplo a placa de som do computador).

A tabela 2.2 mostra a matriz de informação *MIDI* de um exemplo musical, representado na figura 2.5. Essa informação é necessária para gravar o ficheiro *MIDI* (extensão .mid).

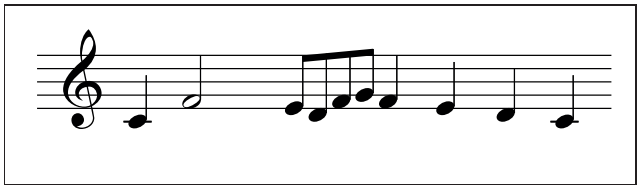

Figura 2.5: Exemplo musical.

Canal	Nota	Intensidade	Tempo de início (s)	Tempo de fim (s)
0	60	127	0.0	1.0
0	65	127	1.0	3.0
0	64	127	3.0	3.5
0	62	127	3.5	4.0
0	65	127	4.0	4.5
0	67	127	4.5	5.0
0	65	127	5.0	6.0
0	64	127	6.0	7.0
0	62	127	7.0	8.0
0	60	127	8.0	9.0

Tabela 2.2: Exemplo de matriz de informação *MIDI* do exemplo musical da figura 2.5. Considera-se a duração da semínima igual a 1 segundo e o valor máximo de intensidade em toda a música. É usado apenas o primeiro canal.

2.3 Lei de Zipf como avaliador musical

A lei de Zipf [Zipf, 1949,Zipf, 1935] é uma lei que descreve vários fenómenos naturais e humanos. A lei de Zipf aparece em vários contextos, por exemplo na frequência de ocorrência das palavras de um livro [Shtrikman, 1994], na distribuição da população em cidades [Hill, 1970], etc. É um caso particular de uma lei de potência expressa por uma função potência (em que a é próximo da unidade):

$$f(x) = \frac{b}{x^a} \text{ , com } a \text{ e } b \text{ constantes reais}$$

A lei de Zipf é então descrita pela forma:

$$f(n) = \frac{K}{n} \text{ , } n = 1, 2, 3, ...$$

em que *f(n)* é a frequência de ocorrência de um determinado objeto ligado à sua ordem n e K é uma constante real. O segundo elemento repete-se com uma frequência que é metade da do primeiro, o terceiro elemento com uma frequência de 1/3 da do primeiro, e assim sucessivamente. Notar que $K = f(1)$.

Considera-se $f_a(n)$ como sendo a frequência de ocorrência absoluta do n-ésimo elemento. Seja N o número de elementos distintos de um dado conjunto de elementos, então a frequência de ocorrência relativa do n-ésimo elemento é:

$$f(n) = \frac{f_a(n)}{\sum_{n=1}^{N} f_a(n)} \text{ , } n = 1, 2, 3, ...$$

Para avaliar o quanto um determinado conjunto de dados se adequa à lei de Zipf seguem-se os seguintes passos:

1. ordena-se (de forma decrescente) os objetos pela sua frequência de ocorrência relativa ou absoluta, respetivamente $f(n)$ ou $f_a(n)$, e atribui-se a ordem n a cada objeto;

2. em vez de se analisar n e $f(n)$, ou n e $f_a(n)$, estuda-se:

$$\tilde{n} = \log_{10}(n) \quad \text{e} \quad \tilde{f}(n) = \log_{10} f(n)$$

ou

$$\tilde{n} = \log_{10}(n) \quad \text{e} \quad \tilde{f}_a(n) = \log_{10} f_a(n)$$

3. com os pontos $(\tilde{n},\tilde{f}(n))$ ou $(\tilde{n},\tilde{f}_a(n))$ constrói-se um diagrama de dispersão;

4. usa-se o método dos mínimos quadrados (regressão linear) obtendo uma reta representativa, tendo assim o valor do declive (m) e do coeficiente de determinação (R^2);

5. no gráfico log-log, assumindo uma distribuição de probabilidades ideal de acordo com a lei de Zipf, o declive será -1 e o coeficiente de determinação será 1. Tendo esses valores como referência, então compara-se os valores obtidos com os ideais. Optou-se por usar a distância euclidiana como valor do erro:

$$erro = \sqrt{(m_{ideal} - m)^2 + (R^2_{ideal} - R^2)^2} =$$

$$= \sqrt{(-1 - m)^2 + (1 - R^2)^2}$$

Neste trabalho a lei de Zipf é usada no contexto musical com o objetivo de avaliar a agradabilidade de uma música. A lei de Zipf analisa a música objetivamente num contexto estatístico. Neste trabalho é apenas analisada a validade da lei de Zipf nas notas e durações. Há vários trabalhos [Lo, 2012, Manaris et al., 2003, Manaris et al., 2005, Zanette, 2006] que evidenciaram propriedades da lei de Zipf em várias músicas bem conhecidas. Assim, considerou-se que se as notas ou durações musicais se aproximarem de uma distribuição segundo a lei de Zipf a música tende a ser agradável de ouvir.

Foi definida a região crítica para a propriedade da lei de Zipf como no trabalho de [Lo, 2012]. Se o valor do declive do gráfico log-log de $(\tilde{n},\tilde{f}(n))$ ou $(\tilde{n},\tilde{f}_a(n))$ está entre -0.8 e -1.2, e o coeficiente de determinação (R^2) é maior ou igual a 0.7, então considera-se que os dados seguem uma distribuição segundo a lei de Zipf.

Na tabela 2.3 são mostrados alguns exemplos de valores ideais das frequências de ocorrência relativas seguindo a lei de Zipf para vários conjuntos de N elementos distintos.

Um exemplo da aplicabilidade da lei de Zipf usando notas musicais é apresentado na tabela 2.4, em que o declive do gráfico log-log é -0.49 e o coeficiente de determinação R^2 é 0.93 (ver figura 2.6, $erro = 0.51$), portanto pode-se concluir que este determinado conjunto de notas musicais não se ajusta às propriedades da lei de Zipf, uma vez que o declive não está no intervalo desejado $[-1.2, -0.8]$.

Número de elementos distintos	2	3	4	5
$f(1)$	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{12}{25}$	$\frac{60}{137}$
$f(2)$	$\frac{1}{3}$	$\frac{3}{11}$	$\frac{6}{25}$	$\frac{30}{137}$
$f(3)$	-	$\frac{2}{11}$	$\frac{4}{25}$	$\frac{20}{137}$
$f(4)$	-	-	$\frac{3}{25}$	$\frac{15}{137}$
$f(5)$	-	-	-	$\frac{12}{137}$

Tabela 2.3: Exemplos que têm propriedade da lei de Zipf ideal.

Ordem (n)	Nota musical	Frequência absoluta ($f_a(n)$)	$\tilde{n}$	$\tilde{f}_a(n)$
1	60	48	0.0000	1.6812
2	67	45	0.3010	1.6532
3	65	28	0.4771	1.4472
4	62	27	0.6021	1.4314
5	64	23	0.6990	1.3617
6	69	22	0.7782	1.3424
7	70	20	0.8451	1.3010

Tabela 2.4: Exemplo com indicação das frequências de ocorrência de um conjunto de notas musicais.

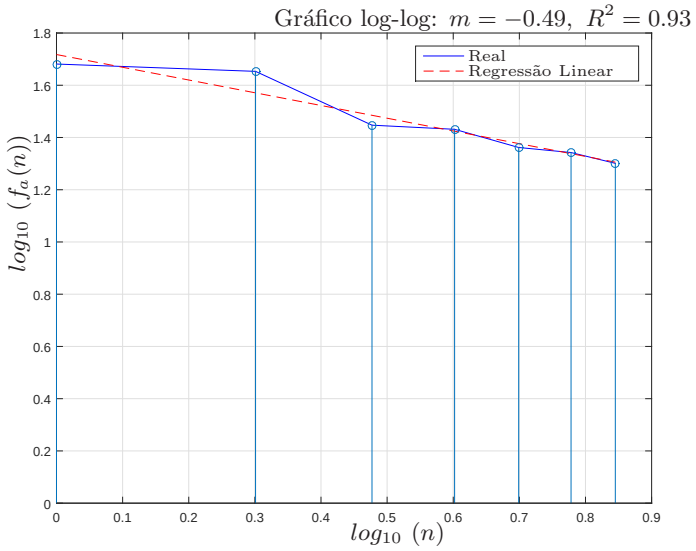

Figura 2.6: Gráfico log-log da aplicação da lei de Zipf usando o exemplo da tabela 2.4.

O afastamento das frequências de ocorrências dos elementos de um determinado conjunto de dados em relação à distribuição ideal de probabilidades segundo a lei de Zipf é o avaliador musical apresentado neste trabalho. Apresentou-se neste capítulo como a informação musical é guardada, e como a música é avaliada. Em seguida, é apresentado como é feita a conversão da sequência simbólica para uma sequência numérica.

Capítulo 3

Do ADN aos números

Neste capítulo é explicado como a informação do ADN é tratada, e como é convertida em números. Através da ferramenta desenvolvida neste trabalho é possível converter qualquer sequência simbólica de ADN em informação numérica, para depois ser convertida em música.

3.1 Conversão de oligonucleótidos, conversão em aminoácidos e conversão *ECG*

Considera-se um oligonucleótido como uma subsequência de N-nucleótidos consecutivos (neste trabalho considerou-se $N \in \{1, 2, 3, 4\}$), ou seja um oligonucleótido pode referir-se a um nucleótido ($N = 1$), di-nucleótido ($N = 2$), tri-nucleótido ($N = 3$), ou ainda tetra-nucleótido ($N = 4$), N corresponde ao tamanho da palavra. A conversão de tri-nucleótidos em aminoácidos é mostrada na tabela 3.1.

A conversão em grupos de composição equivalente (*ECG - Equivalent Composition Group*, consultar [Afreixo et al., 2014]) pode ser realizada em palavras de qualquer tamanho, esta conversão não pode ser realizada simultaneamente com a conversão para aminoácidos. Para esta atribuição considera-se que os nucleótidos A e T são do "tipo 1", e os nucleótidos C e G são do "tipo 2". Cada grupo *ECG* engloba todas as palavras com o mesmo número de nucleótidos de um determinado tipo. A título de exemplo, considera-se que o grupo 1 possui todas as palavras com 0 nucleótidos do tipo 2, o grupo 2 possui todas as palavras com 1 nucleótido do tipo 2, o grupo n possui todas as palavras com $n - 1$ nucleótidos do tipo 2. Todas as palavras associadas a um grupo são substituídas por o número do grupo, ou seja, passam a ter outro valor (pode ser entendido como uma nova palavra). Na tabela 3.2 é mostrada a atribuição *ECG* para palavras de tamanho 2.

Os oligonucleótidos possuem 4^N palavras distintas (N é o tamanho da palavra), a conversão em aminoácidos produz 21 palavras diferentes, e a conversão *ECG* produz $N + 1$ conjuntos distintos.

Nos nucleótidos fez-se a seguinte atribuição numérica: $A \to 1$; $C \to 2$; $G \to 3$; $T \to 4$. Esta atribuição respeita a ordem lexicográfica (também conhecida como ordem alfabética) e da mesma forma é usada nos di-nucleótidos, tri-nucleótidos e tetra-nucleótidos (no caso dos aminoácidos não foi respeitada a ordem lexicográfica, ver tabela 3.1). Por exemplo no caso

dos di-nucleótidos tem-se:

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

De notar que na aplicação desenvolvida neste trabalho, se o tamanho da sequência a converter não for múltiplo do tamanho da palavra (N), então os últimos nucleótidos são ignorados.

3.2 Distância entre palavras

A distância entre palavras [Nair and Mahalakshmi, 2005, Afreixo et al., 2009] é um mapeamento alternativo que permite guardar a informação genética (A, C, G e T). Este mapeamento permite uma representação numérica, o que é útil para análise de dados.
As distâncias entre palavras de uma sequência genómica são representadas através de vetores numéricos. A distância entre palavras corresponde ao número de posições que estão entre uma determinada palavra (exclusive) e a próxima palavra igual (inclusive). Partindo do(s) vetor(es) numérico(s) das distâncias é possível recuperar a sequência que contém as palavras, admitindo que as posições iniciais de cada palavra são conhecidas. É possível fazer um vetor numérico para cada palavra distinta (por exemplo no caso de aminoácidos, tem-se 21 vetores numéricos), ou pode-se apenas fazer um vetor numérico com as distâncias de todas as palavras. Quando não é encontrada uma próxima palavra igual na sequência podem ser consideradas duas técnicas, ou se volta ao início da sequência à procura da próxima palavra igual (técnica circular) ou se considera uma posição à frente do fim da sequência como a próxima palavra igual (técnica não-circular, usada neste trabalho).

De seguida é apresentado um exemplo, considera-se o tamanho da palavra unitário ($N = 1$), com a seguinte sequência simbólica:

$$S_s = \text{`}ACCGTATG\text{'}$$

Convertendo os nucleótidos para números, a sequência numérica é, então:

$$S_n = [1 \quad 2 \quad 2 \quad 3 \quad 4 \quad 1 \quad 4 \quad 3]$$

A distância do primeiro 'A' ao segundo 'A' é 5, a distância do primeiro 'C' ao segundo 'C' é 1, etc. Ficando o vetor numérico das distâncias então (técnica não-circular):

$$d = \{5, 1, 6, 4, 2, 3, 2, 1\}$$

Considerando quatro vetores numéricos individuais, um para cada nucleótido, pode-se chegar facilmente ao resultado:

$$d_A = \{5, 3\}$$

$$d_C = \{1, 6\}$$

$$d_G = \{4, 1\}$$

$$d_T = \{2, 2\}$$

Codão	Nome	Número	Codão	Nome	Número
AAA	Lisina	14	GAA	Ácido glutâmico	16
AAC	Asparagina	13	GAC	Ácido aspártico	15
AAG	Lisina	14	GAG	Ácido glutâmico	16
AAT	Asparagina	13	GAT	Ácido aspártico	15
ACA	Treonina	8	GCA	Alanina	9
ACC	Treonina	8	GCC	Alanina	9
ACG	Treonina	8	GCG	Alanina	9
ACT	Treonina	8	GCT	Alanina	9
AGA	Arginina	19	GGA	Glicina	20
AGC	Serina	6	GGC	Glicina	20
AGG	Arginina	19	GGG	Glicina	20
AGT	Serina	6	GGT	Glicina	20
ATA	Isoleucina	3	GTA	Valina	5
ATC	Isoleucina	3	GTC	Valina	5
ATG	Metionina, *Start*	4	GTG	Valina, *Start*	5
ATT	Isoleucina, *Start*	3	GTT	Valina	5
CAA	Glutamina	12	TAA	*Stop*	21
CAC	Histidina	11	TAC	Tirosina	10
CAG	Glutamina	12	TAG	*Stop*	21
CAT	Histidina	11	TAT	Tirosina	10
CCA	Prolina	7	TCA	Serina	6
CCC	Prolina	7	TCC	Serina	6
CCG	Prolina	7	TCG	Serina	6
CCT	Prolina	7	TCT	Serina	6
CGA	Arginina	19	TGA	*Stop*	21
CGC	Arginina	19	TGC	Cisteína	17
CGG	Arginina	19	TGG	Triptofano	18
CGT	Arginina	19	TGT	Cisteína	17
CTA	Leucina	2	TTA	Leucina	2
CTC	Leucina	2	TTC	Fenilalanina	1
CTG	Leucina, *Start*	2	TTG	Leucina, *Start*	2
CTT	Leucina	2	TTT	Fenilalanina	1

Tabela 3.1: Tabela do código genético.

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
1	2	2	1	2	3	3	2	2	3	3	2	1	2	2	1

Tabela 3.2: Tabela de conversão *ECG* no caso de di-nucleótidos (tamanho da palavra = 2).

É possível calcular as distâncias de sequências com qualquer tipo de palavras (di-nucleótidos, tri-nucleótidos, tetra-nucleótidos, aminoácidos ou palavras ECG). Para palavras com pelo menos dois nucleótidos as distâncias podem ser determinadas com ou sem janela deslizante. No caso de não se usar janela deslizante existem N *reading frames*. A janela deslizante introduz redundância no vetor numérico. A sequência simbólica pode ser obtida através do(s) vetor(es) das distâncias entre palavras, admitindo que são conhecidas *a priori* as posições inicias de todas as palavras distintas.

Em seguida, apresenta-se um exemplo ilustrativo do uso de janela deslizante com tamanho da palavra dois ($N = 2$), considerando a seguinte sequência simbólica:

$$S_s = \text{`}ACGACTTTTTCCGAAGCGCGAACCGT\text{'}$$

Com janela deslizante a sequência numérica S_n é:

$$[2 \quad 7 \quad 9 \quad 2 \quad 8 \quad 16 \quad 16 \quad 16 \quad 16 \quad 14 \quad 6 \quad 7 \quad 9 \quad 1 \quad 3 \quad 10 \quad 7 \quad 10 \quad 7 \quad 9 \quad 1 \quad 2 \quad 6 \quad 7 \quad 12 \quad 13]$$

Notar que o último valor da sequência numérica (13) diz respeito ao último nucleótido reagrupado com o primeiro nucleótido, ou seja no final da sequência os últimos nucleótidos reagrupam-se com os primeiros.

Sem janela deslizante temos N *reading frames*, neste caso $N = 2$, as duas *reading frames* são:

$$r_1 = [2 \quad 9 \quad 8 \quad 16 \quad 16 \quad 6 \quad 9 \quad 3 \quad 7 \quad 7 \quad 1 \quad 6 \quad 12]$$

$$r_2 = [7 \quad 2 \quad 16 \quad 16 \quad 14 \quad 7 \quad 1 \quad 10 \quad 10 \quad 9 \quad 2 \quad 7 \quad 13]$$

De notar que o conteúdo das *reading frames* está intercalado na sequência numérica determinada com janela deslizante.

Os resultados das distâncias são bastante diferentes com e sem janela deslizante (foi usada a técnica não-circular):

- com janela deslizante:

$$d = \{3, 10, 10, 18, 22, 1, 1, 1, 18, 17, 12, 5, 7, 7, 12, 2, 2, 9, 5, 7, 6, 5, 4, 3, 2, 1\}$$

- sem janela deslizante:

$$d_{r_1} = \{13, 5, 11, 1, 9, 6, 7, 6, 1, 4, 3, 2, 1\}$$

$$d_{r_2} = \{5, 9, 1, 10, 9, 6, 7, 1, 5, 4, 3, 2, 1\}$$

Assim, os dezasseis $\left(4^2\right)$ vetores numéricos individuais são (apenas foram apresentados para o caso do uso de janela deslizante):

$d_{AA} = \{7, 6\}$	$d_{AC} = \{3, 18, 5\}$	$d_{AG} = \{12\}$	$d_{AT} = \{\}$
$d_{CA} = \{\}$	$d_{CC} = \{12, 4\}$	$d_{CG} = \{10, 5, 2, 5, 3\}$	$d_{CT} = \{22\}$
$d_{GA} = \{10, 7, 7\}$	$d_{GC} = \{2, 9\}$	$d_{GG} = \{\}$	$d_{GT} = \{2\}$
$d_{TA} = \{1\}$	$d_{TC} = \{17\}$	$d_{TG} = \{\}$	$d_{TT} = \{1, 1, 1, 18\}$

3.3 Frequência de ocorrência das palavras

A frequência de ocorrência das palavras é outra caraterística muito importante usado neste trabalho para a conversão musical. Este mapeamento consiste em atribuir a cada palavra um número de ordem relativo à sua frequência de ocorrência. É atribuído o número de ordem 1 à palavra mais provável, o número de ordem 2 à segunda palavra mais provável, etc. No caso da frequência de ocorrência de duas ou mais palavras ser igual, então as palavras que terão mais prioridade (serão atribuídos números de ordem inferiores) são as palavras a que correspondem menores índices [1]. Este mapeamento pode ser usado em oligonucleótidos, aminoácidos ou palavras *ECG*.

Tomando como exemplo a sequência de nucleótidos ($N = 1$), $S_s =$ '$GGCCTTCTAA$'.
Então a frequência de ocorrência de cada nucleótido é:

$$f_C = f_T = \frac{3}{10} \quad > \quad f_A = f_G = \frac{2}{10}$$

Notar que neste exemplo, há casos de igualdade de frequências de ocorrência entre nucleótidos, pelo que é preciso respeitar a ordem lexicográfica para atribuir as ordens:

$$C \rightarrow 1, \quad T \rightarrow 2, \quad A \rightarrow 3, \quad G \rightarrow 4.$$

Finalmente para obter o vetor numérico respetivo às ordens das frequências de ocorrência, é só necessário substituir cada nucleótido pela ordem correspondente:

$$S_o = [4 \quad 4 \quad 1 \quad 1 \quad 2 \quad 2 \quad 1 \quad 2 \quad 3 \quad 3]$$

Outro exemplo com conversão para aminoácidos é apresentado em baixo, considere-se a sequência simbólica:

$$S_s = \text{'}TTTGAAC\text{'}$$

Usando janela deslizante (notar que os últimos nucleótidos são reagrupados com os primeiros), então a sequência numérica respetiva é:

$$S_n = [1 \quad 2 \quad 21 \quad 16 \quad 13 \quad 8 \quad 2]$$

Resulta então (respeitando as prioridades), o vetor numérico com os número de ordens das frequências de ocorrência:

$$S_o = [2 \quad 1 \quad 6 \quad 5 \quad 4 \quad 3 \quad 1]$$

3.4 Divisão em janelas

A divisão em janelas é um mapeamento alternativo criado neste trabalho, traz uma perspetiva diferente à forma como se interpreta a sequência, que veio na sequência de ter possibilidade de ouvir a música de cada palavra individualmente. A divisão em janelas consiste na divisão

[1] Lembrar que a ordem lexicográfica apenas foi usada no caso dos oligonucleótidos para a atribuição numérica. No caso dos aminoácidos consultar a tabela 3.1. A própria conversão *ECG* já impõe uma atribuição numérica, pelo que não foi necessária fazer outra atribuição. Exemplo: o aminoácido Fenilalanina (número 1) tem uma prioridade superior ao aminoácido Glicina (número 20).

da sequência em janelas de tamanho K (nucleótidos). Foi imposto neste trabalho que o tamanho da janela (K) tenha um valor inteiro compreendido no intervalo [10, 4000], o maior valor possível para o tamanho da janela corresponde a 4000 nucleótidos. Com a divisão em janelas o algoritmo não permite o uso de janela deslizante.

É apresentado um exemplo com tamanho da palavra unitário ($N = 1$) e tamanho da janela $K = 10$, de sequência simbólica $S_s =$ '$ACCGGGTTTTAAAACCCGGT$'. Depois da divisão em janelas a sequência fica com o aspeto:

[Início - *Start*] [ACCGGGTTTT] [AAAACCCGGT] [Fim - *End*]

Neste exemplo, temos apenas duas janelas (de tamanho $K = 10$ nucleótidos).

De notar, que se o tamanho da sequência não for múltiplo do tamanho da janela (K), os últimos nucleótidos são ignorados. É obrigatório que o tamanho da janela seja um valor múltiplo do tamanho da palavra.
O conceito das distâncias entre palavras e os números de ordem das frequências de ocorrência é neste caso calculado de forma diferente, pelo que será apresentada em seguida como é feita essa conversão.

3.4.1 Distância entre palavras no caso de divisão em janelas

Neste caso as distâncias são determinadas apenas para cada palavra individualmente, ou seja no final há vários vetores numéricos de distância (cada um relativo a uma determinada palavra). A contagem começa na primeira ocorrência de uma determinada palavra, após o fim da janela é encontrada a próxima ocorrência da mesma palavra, essa distância corresponde ao primeiro valor do vetor de distâncias associado a uma determinada palavra. Depois, a distância é calculada a partir dessa ocorrência até à primeira ocorrência da palavra após o fim da janela, e assim sucessivamente até que quando chega ao fim, é calculada a distância tendo em conta o fim "imaginário" (*End*), equivalente à técnica não-circular.

Usando o exemplo anterior ($N = 1$, $K = 10$), $S_s =$ '$ACCGGGTTTTAAAACCCGGT$':

[Início - *Start*] [ACCGGGTTTT] [AAAACCCGGT] [Fim - *End*]

Os vetores de distâncias dos nucleótidos são:

$$d_A = \{10, 10\}$$
$$d_C = \{13, 6\}$$
$$d_G = \{14, 3\}$$
$$d_T = \{13, 1\}$$

3.4.2 Frequência de ocorrência das palavras no caso de divisão em janelas

Da mesma maneira apresentada anteriormente uma determinada sequência é divida em janelas de tamanho K (nucleótidos), e é atribuído um número de ordem a cada palavra consoante a sua frequência de ocorrência. A palavra mais frequente é correspondida pelo número de

ordem 1, a segunda palavra mais frequente corresponde ao número de ordem 2, etc. Números de ordem inferiores correspondem a frequências de ocorrência superiores. No caso de existirem frequências de ocorrência iguais entre palavras, procede-se de maneira semelhante ao caso de não existir a divisão em janelas, dá-se prioridade a palavras que correspondem a índices inferiores. É construído para cada palavra um vetor que contém os números de ordem relacionadas com a sua frequência de ocorrência em cada janela, ou seja, para cada janela é feita uma nova correspondência entre os números de ordem e as palavras.

Com a mesma sequência ($N = 1,\ K = 10$), $S_s =$ '$ACCGGGTTTTAAAACCCGGT$':

[Início - *Start*] [ACCGGGTTTT] [AAAACCCGGT] [Fim - *End*]

Os vetores numéricos finais respetivos aos números de ordem das frequências de ocorrência de cada palavra são então:

$$S_{o_A} = [4 \quad 1]$$
$$S_{o_C} = [3 \quad 2]$$
$$S_{o_G} = [2 \quad 3]$$
$$S_{o_T} = [1 \quad 4]$$

Uma vez que na primeira janela a ordem das frequências de ocorrência é:

$$f_T > f_G > f_C > f_A$$

A atribuição dos números de ordem é a seguinte: $T \to 1,\ G \to 2,\ C \to 3,\ A \to 4$.

E na segunda janela a ordem das frequências de ocorrência é:

$$f_A > f_C > f_G > f_T$$

A atribuição dos números de ordem é a seguinte: $A \to 1,\ C \to 2,\ G \to 3,\ T \to 4$.

3.5 Distribuição das distâncias entre palavras

No sentido de explicar as distribuições das distâncias tanto no caso em que não se usa divisão em janelas como no caso em que se usa divisão em janelas, assume-se um contexto de independência entre palavras, portanto sem o uso de janela deslizante.

No caso em que não se usa a divisão em janelas, a distribuição é geométrica. Define-se D a variável aleatória que representa a distância entre a ocorrência de duas palavras consecutivas, e p a probabilidade da palavra em estudo:

$$P(D = d) = p(1-p)^{d-1} \quad , \quad d = 1, 2, ...$$

Usando divisão em janelas, define-se D a variável aleatória que representa a distância entre a primeira ocorrência da palavra w numa janela e a primeira ocorrência de w após o fim da janela, seja p a probabilidade da palavra w e K o tamanho da janela, então:

$$P(D = d) = p^2 \sum_{i=1}^{min\{d,K\}} (1-p)^{|K-d|+2(i-1)} \quad , \quad d = 1, 2, ...$$

Esta distribuição de distâncias é diferente da anterior, e tem o seu valor máximo para a distância igual ao tamanho da janela (ver figura 3.1 para comparar as distribuição das distâncias com e sem divisão em janelas).

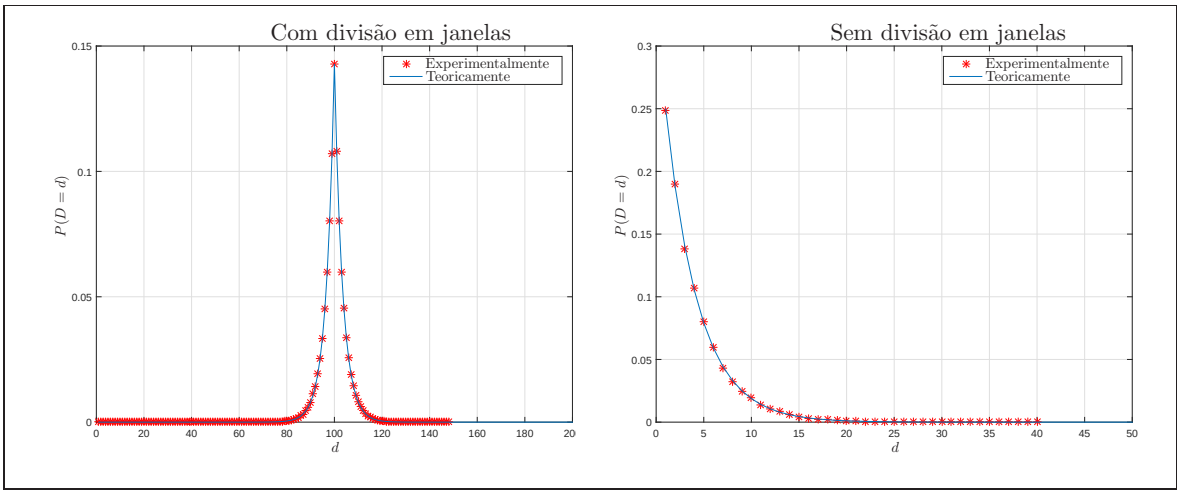

Figura 3.1: Distribuição das distâncias com e sem divisão em janelas: foi usado tamanho da palavra um (nucleótidos) em ambos os casos, e admitiu-se probabilidades de ocorrência independentes e equiprováveis (25%). Foi usado tamanho da janela 100 no caso de divisão em janelas.

Capítulo 4

Dos números à música

Este capítulo esclarece como a informação numérica, pré-convertida de sequências de ADN, é usada para criar música. É explicado como são determinadas as notas, durações e intensidades musicais. As notas e as durações podem ser atribuídas através dos números de ordem das frequências de ocorrência das palavras ou das distâncias entre palavras. As intensidades musicais são calculadas a partir de outras funções que estão relacionadas com as frequências das palavras numa determinada janela. São apresentados exemplos completos de conversão para música. É usada a lei de Zipf de forma a criar música mais agradável na perspetiva do ouvinte. A ferramenta desenvolvida neste trabalho possui um bloco que é responsável pela conversão de sequências simbólicas de ADN em informação musical (ou seja, primeiro faz a conversão do ADN aos números, que foi apresentada no capítulo anterior, e por último faz a conversão dos números à música, que é apresentado neste capítulo).

4.1 Atribuição das notas e durações musicais

4.1.1 Notas musicais

No programa desenvolvido neste trabalho, uma nota musical pode ter um valor inteiro no intervalo $[0, 127]$, em que foi imposto que o número "0" corresponde à pausa, momento de silêncio ("60" corresponde ao Dó central, para ver os restantes números correspondentes às notas consultar a tabela 2.1 do capítulo 2). Foi também imposto que o número máximo de notas distintas a usar num trecho musical são 32, notar que nem sempre são usadas todas as notas.

Notas musicais através das frequências de ocorrência

Numa determinada música as notas a usar são um parâmetro de entrada da aplicação, pelo que o utilizador pode escolher quais notas deseja usar na composição da música. Através do vetor das notas desejadas pelo utilizador e dos números de ordem das frequências de ocorrência das palavras, são determinadas as notas a usar. Nos casos em que o número de valores distintos dos números de ordem excedem o número de notas desejadas, então é necessário fazer um reagrupamento dos números de ordem. Nesse caso optou-se por se realizar uma otimização segundo a lei de Zipf (idealmente respeita a distribuição de probabilidades segundo a lei de Zipf, mais informação nas secções 2.3 e 4.4).

Finalmente, depois de ter o vetor final com os números de ordem (N_o), é apenas necessário substituir os números de ordem pelas notas desejadas (correspondentes àquela ordem).

De seguida um exemplo é mostrado, admita-se o seguinte vetor de números de ordem:

$$N_o = [1 \quad 4 \quad 8 \quad 5 \quad 2 \quad 7 \quad 2 \quad 5 \quad 3 \quad 4 \quad 6]$$

E as notas musicais desejadas (por ordem de preferência) são neste caso oito:

$$\text{notas desejadas} = [60 \quad 62 \quad 64 \quad 65 \quad 67 \quad 69 \quad 71 \quad 72]$$

Então o número 1 do vetor N_o (que corresponde à(s) palavra(s) mais provável(eis)) será substituído pela primeira nota desejada (neste caso 60), o número 2 do vetor N_o (que corresponde à(s) segunda(s) palavra(s) mais provável(eis)) será substituído pela segunda nota desejada (62), e assim sucessivamente. O vetor das notas musicais constitui em parte a pauta musical (substituir cada número do vetor dos números de ordem N_o pela nota respetiva):

$$\text{notas} = [60 \quad 65 \quad 72 \quad 67 \quad 62 \quad 71 \quad 62 \quad 67 \quad 64 \quad 65 \quad 69]$$

A melodia resultante está representada na figura 4.1 (neste exemplo são ignoradas as durações e a dinâmica musical).

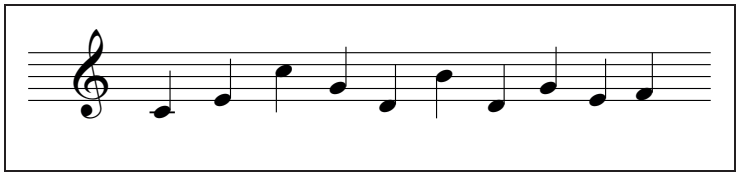

Figura 4.1: Exemplo de conversão dos números de ordem relativos às frequências de ocorrência em notas musicais.

Notas musicais através das distâncias entre palavras

O mapeamento neste caso é idêntico ao do das frequências de ocorrência, no caso de existir um número de distâncias distintas superior ao número desejado de notas a usar, então é necessário fazer um reagrupamento de distâncias (de acordo com a otimização de Zipf, ver secção 4.4). À classe de distâncias número 1 corresponde a nota nº 1, à classe de distâncias número 2 corresponde a nota nº 2, etc. Uma vez que as distâncias entre palavras têm normalmente vários valores distintos então é muito provável que a otimização de Zipf seja realizada.

4.1.2 Durações musicais

Para a criação de uma música, foi imposto o uso máximo de 8 durações distintas, durações essas que podem ter um de 19 valores diferentes (podem ser subentendidos como uma figura musical):

	Fusa	Semicolcheia	Colcheia	Semínima	Mínima	Semibreve	Breve
Sem ponto	0.125	0.25	0.5	1	2	4	8
.	0.1875	0.375	0.75	1.5	3	6	-
..	0.21875	0.4375	0.875	1.75	3.5	7	-

Lembrar que um ponto de aumentação à frente de uma figura musical faz com que a duração seja 1.5 vezes superior e dois pontos de aumentação à frente faz com que a duração seja 1.75 vezes superior.

A atribuição das durações é realizada da mesma forma que a atribuição das notas: pode ser determinada através dos números de ordem ou das distâncias entre palavras. Através da interface o utilizador especifica quais as durações desejadas (no máximo 8). Devido a serem usadas no máximo 8 durações distintas, será necessário mais vezes a realização da otimização de Zipf (reagrupamento de dados).

De seguida é apresentado um exemplo. Admitindo que as durações são determinadas a partir dos números de ordem relativos às frequências de ocorrência, cujo vetor dos números de ordem é:

$$N_o = [1 \quad 4 \quad 8 \quad 5 \quad 2 \quad 7 \quad 2 \quad 5 \quad 3 \quad 4 \quad 6]$$

E as durações musicais desejadas (por ordem de preferência) são neste caso oito:

$$\text{durações desejadas} = [0.5 \quad 0.75 \quad 1 \quad 1.5 \quad 2 \quad 3 \quad 4 \quad 6]$$

Então é necessário criar o vetor das durações relativo à pauta musical, substituindo os valores do vetor N_o correspondentes às durações desejadas, o número 1 do vetor N_o corresponde ao primeiro valor do vetor das durações desejadas, neste caso é 0.5, o número 2 do vetor N_o corresponde ao segundo valor do vetor das durações desejadas, neste caso é 0.75, e assim sucessivamente. O vetor final das durações é então:

$$\text{durações} = [0.5 \quad 1.5 \quad 6 \quad 2 \quad 0.75 \quad 4 \quad 0.75 \quad 2 \quad 1 \quad 1.5 \quad 3]$$

As durações resultantes estão apresentadas na figura 4.2 (neste exemplo são ignoradas as notas musicais e dinâmica musical).

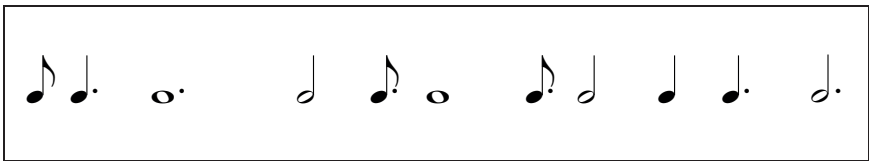

Figura 4.2: Exemplo de conversão dos números de ordem em durações.

4.2 O cálculo das intensidades (dinâmica musical)

No programa desenvolvido deste trabalho, a cada intensidade atribui-se de forma empírica um valor inteiro no intervalo [64, 127] (devido ao padrão MIDI, mais informação na secção 2.2 do capítulo 2), em que "64" corresponde à menor intensidade possível e "127" corresponde à maior intensidade possível. Há uma exceção, no caso das pausas (silêncio) a intensidade é "0".

O cálculo de cada intensidade está relacionado com a frequência de ocorrência de uma palavra numa determinada janela. Foram implementadas dois métodos diferentes, um para o caso em que não há divisão em janelas, e outro para o caso em que há divisão em janelas.

4.2.1 Intensidades no caso em que não há divisão em janelas

No caso em que não há divisão em janelas, é usada uma janela de tamanho $4 \times D$, em que D corresponde ao número de palavras distintas (consoante o tamanho da palavra, conversão para aminoácidos ou conversão para ECG), por exemplo no caso dos aminoácidos, $D = 21$. O tamanho desta janela foi atribuído de forma empírica.

Para cada palavra é determinada a sua frequência de ocorrência numa determinada janela, essa frequência de ocorrência é proporcional ao valor da intensidade musical.

Existe dois casos possíveis:

- sequência de tamanho inferior ou igual a $4D$: neste caso as frequências de ocorrência de cada palavra são determinadas na sequência toda;
- sequência de tamanho superior a $4D$: as frequências de ocorrência de cada palavra são determinadas numa janela de tamanho $4D$ que se inicia na palavra para o qual está a ser calculada. Para as últimas $4D$ palavras a janela utilizada engloba apenas essas palavras.

Depois de determinadas todas as frequências de ocorrências para todas as palavras, então é feito um reajustamento linear para a gama desejada, [64, 127]. A frequência de ocorrência de uma palavra w numa janela de tamanho $4D$ é dada por:

$$f(w) = \frac{\textit{número de vezes que aparece } w}{4D}$$

De seguida é apresentado um exemplo, assume-se palavras de tamanho 1 (nucleótidos), portanto a janela tem tamanho 16 ($D = 4$). Considera-se a seguinte sequência simbólica de nucleótidos:

$$S = \text{`}ATGTCC\text{'}$$

Neste exemplo, como a sequência tem tamanho inferior a 16, então as frequências de ocorrência são calculadas para toda a sequência:

$$f_A = \frac{1}{6} \qquad f_C = \frac{2}{6} \qquad f_G = \frac{1}{6} \qquad f_T = \frac{2}{6}$$

Após o reajustamento linear para o intervalo [64, 127] tem-se finalmente as intensidades de todas as notas:

$$\text{intensidades} = [64 \quad 127 \quad 64 \quad 127 \quad 127 \quad 127]$$

4.2.2 Intensidades no caso em que há divisão em janelas

No caso em que há divisão em janelas a intensidade é proporcional à diferença entre a frequência de ocorrência local (na janela) de uma palavra e a frequência de ocorrência global (na sequência toda) da mesma palavra. O tamanho da janela usado é o mesmo que no caso da determinação dos números de ordem ou das distâncias entre palavras. A fórmula que calcula essa intensidade vem em função do grau de afastamento da frequência local da palavra (f_L) à frequência global da palavra (f_G):

$$\text{intensidade} = \left| \frac{f_L - f_G}{max(f_L, f_G)} \right|$$

Esta fórmula foi criada de forma empírica, quando o valor da intensidade é próximo de zero significa que a frequência local e a frequência global são semelhantes, quando o valor da intensidade é próximo da unidade significa que a frequência local e a frequência global diferem bastante. Por fim é necessário fazer o reajustamento linear de todos os vetores de intensidades para o intervalo [64, 127].

Um exemplo a seguir é mostrado, assume-se palavras de tamanho 1 (nucleótidos, $N = 1$), e tamanho da janela $K = 10$, a sequência de nucleótidos é a seguinte:

$$[Start]\ [AACTGCCCAG]\ [ATATGTCCCA]\ [End]$$

Em toda a sequência as frequências de ocorrência globais (f_G) são:

$$f_{G_A} = \frac{6}{20}, \quad f_{G_C} = \frac{7}{20}, \quad f_{G_G} = \frac{3}{20}, \quad f_{G_T} = \frac{4}{20}$$

Na primeira janela as frequências de ocorrência locais (f_{WL}) são:

$$f_{L_A} = \frac{3}{10}, \quad f_{L_C} = \frac{4}{10}, \quad f_{L_G} = \frac{2}{10}, \quad f_{L_T} = \frac{1}{10}$$

Na segunda janela as frequências de ocorrência locais (f_{WL}) são:

$$f_{L_A} = \frac{3}{10}, \quad f_{L_C} = \frac{3}{10}, \quad f_{L_G} = \frac{1}{10}, \quad f_{L_T} = \frac{3}{10}$$

Finalmente, as intensidades associadas a cada palavra em cada janela são:

- na primeira janela:
$$i_A = 0, \quad i_C = \frac{1}{8}, \quad i_G = \frac{1}{4}, \quad i_T = \frac{1}{2}$$
- na segunda janela:
$$i_A = 0, \quad i_C = \frac{1}{7}, \quad i_G = \frac{1}{3}, \quad i_T = \frac{1}{3}$$

Neste algoritmo (com divisão em janelas) cada palavra dá origem a uma pauta musical. Após o reajustamento linear de todas as intensidades, as intensidades finais são:

$$\text{intensidades}_{\,\text{A}} = [64 \quad 64]$$

$$\text{intensidades}_{\,\text{C}} = [80 \quad 82]$$

$$\text{intensidades}_{\,\text{G}} = [96 \quad 106]$$

$$\text{intensidades}_{\,\text{T}} = [127 \quad 106]$$

De notar que a maior intensidade ocorre na primeira janela no nucleótido T. O reajustamento linear é feito tendo em conta as intensidades de todas as palavras.

4.3 Exemplo completo de notas, durações e intensidades

Esta secção tem como objetivo descrever um exemplo da junção de todas as variáveis musicais formando assim uma música. É apresentado um exemplo em que não foi usada a divisão em janelas (portanto as intensidades são calculadas em janelas de tamanho $4D$), com o uso de nucleótidos ($N = 1$, técnica não-circular). Considera-se a seguinte sequência simbólica:

$$S = \text{`}CGGCGCATATACGATAAAACCCCCCATGTTG\text{'}$$

As notas musicais foram determinadas através dos números de ordem relativos às frequências de ocorrência e as durações musicais através das distâncias entre palavras.

Os números de ordem das frequências de ocorrência das palavras são:

$$N_o = [2\ \ 3\ \ 3\ \ 2\ \ 3\ \ 2\ \ 1\ \ 4\ \ 1\ \ 4\ \ 1\ \ 2\ \ 3\ \ 1\ \ 4\ \ 1\ \ 1\ \ 1\ \ 1\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 1\ \ 4\ \ 3\ \ 4\ \ 4\ \ 3]$$

As distâncias entre os nucleótidos são:

$$d = \{3, 1, 2, 2, 8, 6, 2, 2, 2, 5, 3, 8, 14, 2, 11, 1, 1, 1, 6, 1, 1, 1, 1, 7, 6, 2, 3, 1, 2, 1\}$$

As intensidades são:

$$\begin{array}{l}[\ 85\ \ 85\ \ 75\ \ 75\ \ 64\ \ 85\ \ 127\ \ 75\ \ 117\ \ 64\ \ 117\ \ 106\ \ 64\ \ 106\ \ 85\ldots \\ \ldots 96\ \ 96\ \ 96\ \ 96\ \ 96\ \ 96\ \ 96\ \ 96\ \ 96\ \ 96\ \ 85\ \ 64\ \ 85\ \ 85\ \ 64\ ]\end{array}$$

Assumindo que:

- as notas desejadas são: [60 62 64 65];
- as durações desejadas são: [0.50 0.75 1.00 1.50 2.00 3.00 4.00 6.00].

Usando os números de ordem, as notas musicais respetivas são então (consultar numeração MIDI na tabela 2.1 do capítulo 2):

$$\begin{array}{l}[\ 62\ \ 64\ \ 64\ \ 62\ \ 64\ \ 62\ \ 60\ \ 65\ \ 60\ \ 65\ \ 60\ \ 62\ \ 64\ \ 60\ \ 65\ldots \\ \ldots 60\ \ 60\ \ 60\ \ 60\ \ 62\ \ 62\ \ 62\ \ 62\ \ 62\ \ 60\ \ 65\ \ 64\ \ 65\ \ 65\ \ 64\ ]\end{array}$$

Devido ao número das distâncias distintas ser superior a oito, algumas distâncias foram reagrupadas (de acordo com a função de otimização de Zipf, ver secção 4.4) ficando o novo vetor de distâncias (d_z) com o seguinte aspeto:

$$d_z = \{3, 1, 2, 2, 6, 4, 2, 2, 2, 3, 3, 6, 8, 2, 7, 1, 1, 1, 4, 1, 1, 1, 1, 5, 4, 2, 3, 1, 2, 1\}$$

Usando as novas distâncias reagrupadas, as durações musicais respetivas são então:

$$\begin{array}{l}[\ 1.00\ \ 0.50\ \ 0.75\ \ 0.75\ \ 3.00\ \ 1.50\ \ 0.75\ \ 0.75\ \ 0.75\ \ 1.00\ \ 1.00\ \ 3.00\ \ 6.00\ \ 0.75\ \ 4.00 \\ 0.50\ \ 0.50\ \ 0.50\ \ 1.50\ \ 0.50\ \ 0.50\ \ 0.50\ \ 0.50\ \ 2.00\ \ 1.50\ \ 0.75\ \ 1.00\ \ 0.50\ \ 0.75\ \ 0.50\ ]\end{array}$$

A pauta musical final é apresentada na figura 4.3 (não foram ilustradas as intensidades).

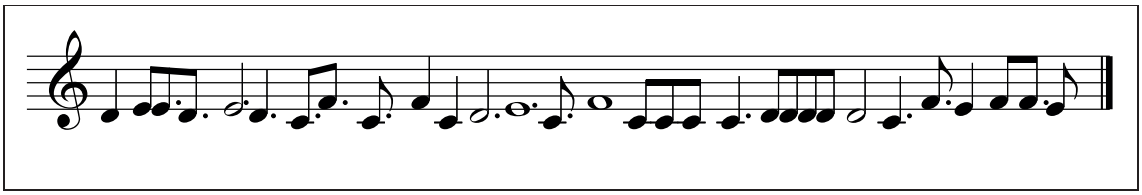

Figura 4.3: Pauta musical correspondente ao exemplo completo da secção 4.3.

4.4 Função de otimização de Zipf (`optZipf`)

Neste trabalho a função de otimização de Zipf tem como objetivo reagrupar dados de uma variável por forma a ajustar a sua distribuição de probabilidades à lei de Zipf (ver secção 2.3 do capítulo 2), em que, o segundo elemento mais provável se repete com uma frequência que é metade da do elemento mais provável, o terceiro elemento com uma frequência que é 1/3 da do elemento mais provável, e assim sucessivamente. Nesta função é imposto que os valores menores de um determinado vetor tenham maiores probabilidades, ou seja, o menor valor, "1", deve ter uma probabilidade aproximadamente do dobro da do segundo menor valor, "2" (ou segundo mais provável), e assim sucessivamente.

A lei de Zipf é descrita pela função:

$$f(n) = \frac{K}{n}$$

em que *f(n)* é a frequência de ocorrência relativa de um determinado objeto ligado à sua ordem n e K é uma constante.

Esta função é usada no algoritmo de conversão quando um determinado vetor (números de ordem ou distâncias entre palavras) tem um número de valores distintos superior ao número de notas/durações que são permitidas usar. Quando esta função é usada nos números de ordem e/ou nas distâncias entre palavras, as notas e/ou as durações musicais tendem a ter distribuições de probabilidades de acordo com a lei de Zipf. É esperado que quando se verifica a lei de Zipf nas notas e/ou durações a música seja mais agradável da perspetiva do ouvinte (resultados do trabalho [Lo, 2012]).

De seguida é explicado como foi implementado o algoritmo da otimização de Zipf, primeiro começa por determinar as probabilidades ideais para um conjunto de T valores distintos, em que T corresponde normalmente ao número máximo de notas ou durações a usar. O conjunto de valores a otimizar, tem de ter um número de valores distintos superior a T, e a otimização só termina quando o número de valores distintos atinge um valor igual a T.

A função começa a verificação das probabilidades nos valores mais baixos (1,2,...), depois determina em qual o caso que minimiza a diferença entre a probabilidade de ocorrência de um valor e a respetiva probabilidade da lei de Zipf ideal:

- no caso em que não é feito nenhum reagrupamento de valores;

ou,

- no caso em que são reagrupados dois ou mais conjuntos de valores consecutivos (por exemplo todos os valores 1 e 2, passam a ser uma única classe: 1).

Não podem ser agrupados conjuntos de valores por forma a que o número de valores distintos do vetor fique inferior a T. No final todos os valores são convertidos para valores inteiros no intervalo [1, T], ou seja, são reagrupados em T classes.

Um exemplo é mostrado de seguida, admitindo as distâncias entre palavras:

$$d = \{3, 1, 2, 2, 2, 2, 2, 2, 7, 4, 2, 8, 2, 5, 2, 5, 1, 4, 4, 5, 3, 9, 9, 2, 2, 2, 2, 2\}$$

Assume-se que as notas serão atribuídas através das distâncias entre palavras, e o máximo número de notas a usar são 4. Pelo que as distâncias terão de ser mapeadas numa de 4 categorias ou classes (usou-se como código de cada categoria os números {1, 2, 3, 4}). Para 4 valores distintos, as probabilidades de ocorrência respeitando a lei de Zipf idealmente são:

$classe$	1	2	3	4
$P(classe)$	0.48	0.24	0.16	0.12

Inicialmente o vetor das distâncias tem 8 valores distintos (1, 2, 3, 4, 5, 7, 8, 9), e as probabilidades de ocorrência de cada distância são:

d (distância)	1	2	3	4	5	7	8	9
$P(d)$	0.0714	0.5000	0.0714	0.1071	0.1071	0.0357	0.0357	0.0714

É desejado que o vetor de distâncias fique apenas com 4 valores distintos, [1, 4], assim começa-se por comparar a probabilidade da distância ser igual a 1 e a probabilidade da distância ser igual a 1 ou 2, com a probabilidade ideal (ver tabela 2.3 do capítulo 2), chega-se à conclusão que a probabilidade conjunta dos números 1 e 2 é a mais próxima da ideal, assim esses dois números passam a ser uma única classe. Iterando sucessivamente até que o vetor de distâncias tenha um número de valores distintos igual a T, chega-se à conclusão que o vetor final das distâncias é:

$$d_z = \{2, 1, 1, 1, 1, 1, 1, 1, 3, 2, 1, 3, 1, 2, 1, 2, 1, 2, 2, 2, 2, 4, 4, 1, 1, 1, 1, 1\}$$

A correspondência entre cada valor de distância e as classes é apresentada na seguinte tabela:

$d(distâncias)$	1	2	3	4	5	7	8	9
$d_z(classes)$	1	1	2	2	2	3	3	4

Através da avaliação da lei de Zipf, verifica-se que o vetor final das distâncias possui um declive $m = -1.65$ e um coeficiente de determinação $R^2 = 0.91$ (portanto este conjunto de valores de distâncias é considerado como não tendo propriedades da lei de Zipf, ver secção 2.3 do capítulo 2).

Capítulo 5

Ferramenta: as suas aplicações e a sua interface gráfica

Este capítulo apresenta a ferramenta desenvolvida neste trabalho que faz uso de todos os algoritmos desenvolvidos (consultar anexo A), que são responsáveis pela conversão do ADN em música e pela avaliação musical de forma objetiva. É exposto um manual da interface, com o objetivo de tornar a aprendizagem pelo utilizador uma tarefa mais simples. Esta aplicação foi desenvolvida em linguagem MATLAB com a ferramenta *Graphical User Interface* (*GUI*), pelo que o programa MATLAB é um requisito necessário à sua utilização. A interface é de um modo geral, de fácil manuseamento para o utilizador, no entanto os conceitos teóricos que estão por detrás das funções realizadas exigem uma leitura pormenorizada desta dissertação. A aplicação permite a conversão de sequências de ADN (a partir de ficheiros do tipo FASTA ou do tipo TEXT do site [EMBL-EBI, 2015]) em música, esta conversão passa primeiro por uma conversão da sequência simbólica em informação musical, que é armazenada num ficheiro de formato .txt. A música é criada através da informação musical, e é armazenada num ficheiro de formato .mid.

5.1 Funcionalidades da aplicação

A aplicação é composta por três principais funcionalidades:

- conversão de ADN em informação musical, que é gravada num ficheiro de extensão .txt;
- conversão da informação musical em música, que é gravada num ficheiro de extensão .mid;
- avaliação musical de forma objetiva: aplicabilidade da lei de Zipf.

5.1.1 Gravação da informação musical num ficheiro de extensão .txt

A informação musical é uma estrutura de dados que contém a informação relativa a uma música. Esta estrutura é composta por:

- número de regiões;
- indicação para cada região, se se trata de uma região código ou não-código, ou se não faz diferenciação;
- número de pautas;
- identificação de todas as notas musicais;
- identificação de todas as durações musicais;
- identificação de todas as intensidades musicais.

Toda esta informação é representada apenas com números num ficheiro do tipo .txt. Inicialmente é gravado na primeira linha do ficheiro o número de regiões, depois para cada região é armazenado um número numa nova linha, para saber se se trata de uma região código ou não-código, ou se não faz diferenciação:

- 0: corresponde a uma região não-código;
- 1: corresponde a uma região código;
- 2: não faz diferenciação entre regiões.

Na próxima linha é gravado o número de pautas que a região contém, de seguida são apresentadas todas as pautas, em que cada pauta é iniciada numa nova linha com o número de notas que contém. Após a indicação do número de notas, é escrita a cada linha uma nova nota, em que em cada linha se escreve o número MIDI da nota, a sua duração e a sua intensidade. Este processo é repetido até não existirem mais regiões.

5.1.2 Criação da música no formato *MIDI* através da informação musical

A aplicação obtém a informação musical a partir da leitura de um ficheiro do tipo .txt consistente. Depois é permitido ao utilizador escolher vários parâmetros:

- escolha das regiões e pautas a ouvir;
- andamento (proporcional à velocidade musical);
- atribuição dos instrumentos para cada pauta musical;
- possibilidade de usar um marcador sonoro para saber se existe distinção entre regiões ou não, e caso haja distingue regiões codificantes e não-codificantes.

A criação do ficheiro MIDI (auxiliadas pelas funções adaptadas de [Schutte, 2015]) consiste em copiar a informação musical do ficheiro do tipo .txt, acrescentando também outras características (instrumentos, andamento, etc.). Como o ficheiro MIDI não armazena a forma de onda sonora resultante de toda a música, mas apenas guarda a informação musical, o ficheiro é relativamente pequeno e a sua criação é rápida.

5.2 Interface gráfica

A interface divide-se em quatro grandes blocos:

- bloco de início (que permite a escolha de um dos próximos três blocos, ver figura 5.1).
- bloco relativo à conversão de ADN em informação musical (a informação musical é gravada num ficheiro do tipo .txt, ver figura 5.2).
- bloco relativo à conversão de informação musical em música (a música é gravada num ficheiro do tipo .mid, ver figura 5.3).
- bloco relativo à avaliação da informação musical usando a lei de Zipf (ver figura 5.4).

Figura 5.1: Bloco de início da interface que possibilita a escolha de três opções diferentes: conversão de ADN em informação musical, criação da música através da informação musical ou ainda avaliação musical dada a informação musical.

5.3 Manual de utilizador

No bloco de conversão de ADN em informação musical, é necessário em primeiro lugar selecionar o ficheiro. Após a seleção do ficheiro (apenas são aceites ficheiros do tipo FASTA ou do tipo TEXT do site [EMBL-EBI, 2015]) é indicado qual o tipo de ficheiro, quantos nucleótidos tem e no caso de fazer distinção entre regiões, indica quantas regiões código e não-código possui. O segundo passo é a escolha de parâmetros (não é obrigatório, uma vez que aparecem parâmetros escolhidos por omissão), pode-se fazer várias atribuições, como por exemplo atribuir os números de ordem das frequências de ocorrência às notas, e as distâncias

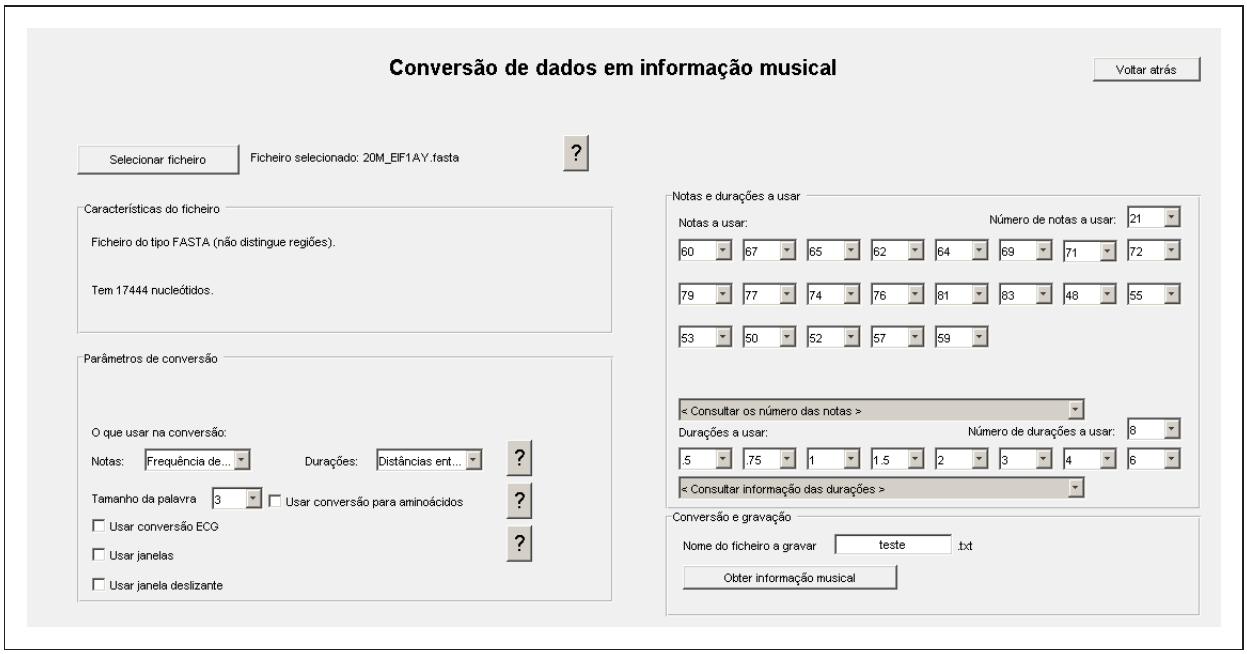

Figura 5.2: Bloco relativo à conversão de ADN em informação musical.

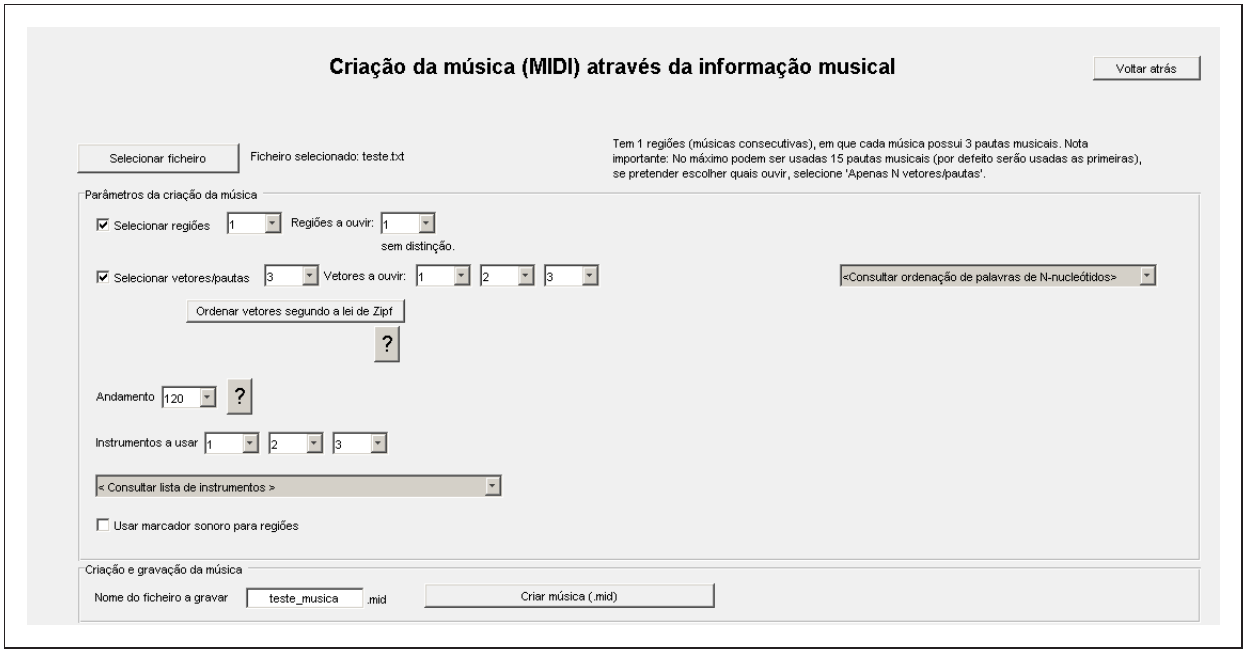

Figura 5.3: Bloco relativo à conversão de informação musical em música.

entre palavras às durações ou vice-versa. Pode-se usar diferentes tamanhos de palavra e usar conversão *ECG*, ou no caso de tri-nucleótidos conversão para aminoácidos. Pode-se dividir a sequência em janelas, nesse caso o algoritmo que determina os números de ordem das frequências de ocorrência, as distâncias entre as palavras e as intensidades é diferente. O número máximo de notas e durações a usar e a indicação de quais notas e durações usar são

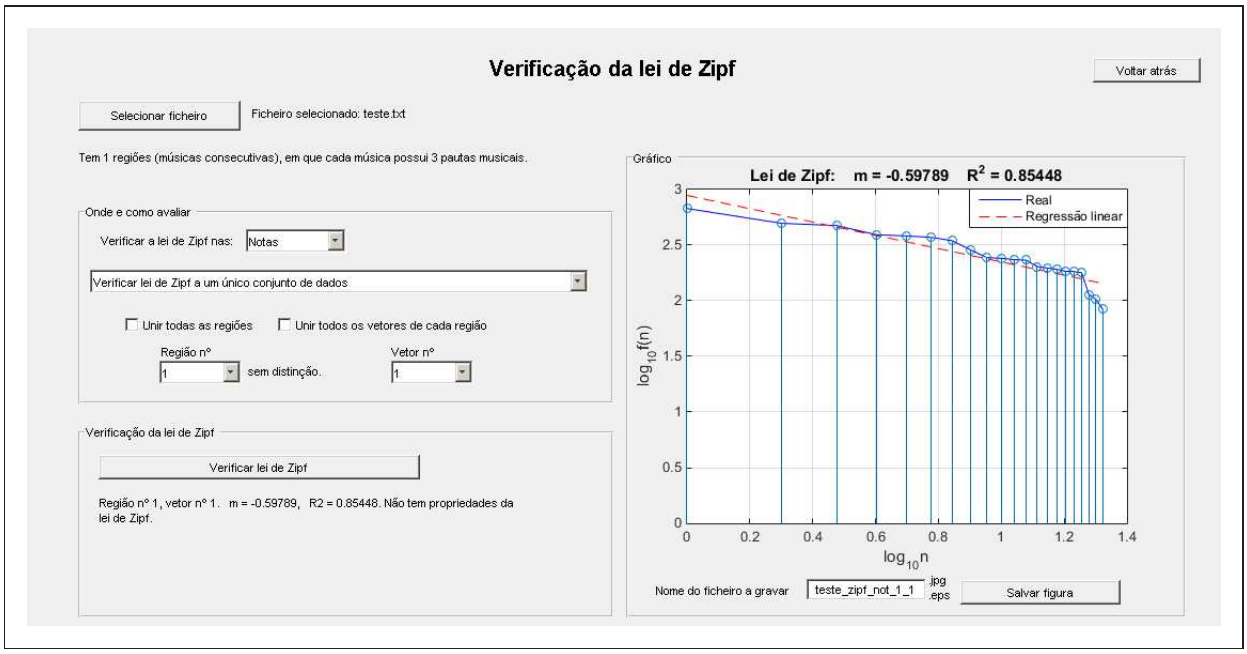

Figura 5.4: Bloco relativo à avaliação objetiva de uma música usando a lei de Zipf.

também parâmetros definidos pelo utilizador. Existem múltiplas combinações, pelo que uma mesma sequência de ADN pode dar origem a várias músicas completamente diferentes.

No bloco respetivo à criação da música (gravada num ficheiro do tipo .mid) através da informação musical o primeiro passo é também a seleção de um ficheiro (do tipo .txt) que contém informação musical (previamente gerado por o bloco de conversão de ADN em informação musical). Após um ficheiro de informação musical estar selecionado, então é permitido ao utilizador definir quantas e quais regiões ouvir, quantas e quais pautas ouvir (no máximo 15, devido à limitação do *General Midi* que apenas permite 15 canais para instrumentos), e no caso de ser necessário excluir algumas pautas então podem ser escolhidas as melhores pautas consoante as propriedades da lei de Zipf. Também é permitido ao utilizador escolher o instrumento para cada pauta musical. O andamento (velocidade da música) é outro parâmetro de entrada. É também possível incluir um marcador sonoro entre regiões (com o objetivo de distinguir regiões código e não-código), este som é realizado no canal que é destinado exclusivamente à percussão, pelo que o som dos marcadores é à base de sons de percussão.

O bloco relativo à avaliação musical permite avaliar a música objetivamente com o uso da lei de Zipf. Novamente o primeiro passo é selecionar o ficheiro que contém a informação musical. De seguida, deve-se escolher se se pretende avaliar o ajustamento da lei de Zipf nas notas ou durações, uma vez que são as duas únicas possibilidades. Após isso, deve-se escolher se se pretende avaliar o ajustamento da lei de Zipf numa determinada região ou pauta, ou se se pretende avaliar em toda a música e mostrar um histograma de quantas pautas têm propriedades da lei de Zipf, quantas pautas possuem apenas o declive correspondente à gama das propriedades da lei de Zipf ($-1.2 \leq m \leq -0.8$), quantas pautas possuem apenas o coeficiente de determinação correspondente à gama das propriedades da lei de Zipf ($R^2 \geq 0.7$), e quantas pautas não possuem nem o declive nem o coeficiente de determinação. É permitido

ao utilizador escolher avaliar todas as pautas ou apenas uma determinada pauta de todas as regiões ou apenas de uma determinada região.

Algumas notas relativas ao programa e à sua interface:

- o número máximo de notas distintas permitidas é 32, e o número máximo de durações distintas permitidas é 8;
- não é possível usar janela deslizante no caso de se usar o algoritmo de divisão da sequência em janelas;
- não é possível usar simultaneamente conversão *ECG* e conversão para aminoácidos;
- é possível usar duas vezes ou mais as mesmas notas/durações;
- para não aumentar a redundância da informação musical, sugere-se que se use sempre os números de ordem das frequências de ocorrência das palavras e as distâncias entre palavras, e não apenas uma delas para a obtenção das notas e durações;
- o tempo estimado máximo de conversão do ADN em informação musical é uma aproximação grosseira (por vezes é um valor bastante sobrestimado).
- no caso de ser desejada apenas uma noção básica sobre o programa, pode-se obter alguma informação clicando nos botões de ajuda que estão na interface.

A ferramenta apresentada tem como objetivo a possibilidade de criação de música genómica de uma maneira simples e eficaz, podendo também avaliar música quanto ao seu ajustamento à lei de Zipf.

Capítulo 6

Resultados

Nos resultados serão apresentados os seguintes tópicos:

- tempos de processamento de conversão de sequências simbólicas de ADN em informação musical usando diferentes parâmetros de entrada no algoritmo desenvolvido neste trabalho;
- distribuição das distâncias entre palavras (com e sem divisão em janelas) antes e depois da aplicação da função de otimização segundo a lei de Zipf (`optZipf`);
- distribuição dos números de ordem das frequências de ocorrência (com e sem divisão em janelas) antes e depois da função de otimização `optZipf`;
- respostas a um formulário realizado a pessoas, e respetivos resultados estatísticos;
- análise comparativa de músicas criadas a partir de regiões código e regiões não-código de diferentes organismos.

6.1 Tempos de processamento de conversão

Para se medirem os tempos de processamento da execução do algoritmo com diferentes parâmetros de conversão, foi usada uma sequência aleatória, uma vez que o tempo da execução do algoritmo depende pouco da sequência usada. A sequência aleatória usada foi criada com probabilidades equiprováveis e tem um total de 60000 nucleótidos. Considerou-se, a título de exemplo, o número máximo de notas a usar e o número máximo de durações a usar 8. Atribuiu-se os números de ordem das frequências de ocorrência às notas musicais e as distâncias entre palavras às durações. Quanto às intensidades não é preciso fazer nenhuma atribuição, pois o seu método de determinação apenas depende se a sequência é dividida em janelas ou não. Os tempos de processamento da conversão da sequência simbólica em informação musical usando diferentes parâmetros são apresentados na tabela tabela 6.1.

No caso de não usar divisão em janelas, cada palavra gera uma nota musical, e usando divisão em janelas em cada janela existem no máximo N notas musicais (em que N é o número de palavras distintas, por exemplo tem-se $N = 21$ para palavras de tamanho 3 e conversão para aminoácidos). Então é intuitivo concluir que o algoritmo que não usa divisão em janelas irá produzir mais notas musicais (consequentemente mais informação musical) e por isso exige

mais tempo de processamento relativamente ao algoritmo que usa divisão em janelas.

Também se verifica que com o aumento do tamanho da palavra o tempo de processamento irá aumentar, uma vez que o número de palavras distintas aumenta. A conversão para grupos *ECG* reduz ligeiramente o tempo de processamento, uma vez que o número de palavras distintas é diminuído.

Com o aumento do tamanho da janela, o número total de notas musicais é diminuído, portanto a informação musical gerada será menor, logo o tempo de processamento diminui.

No caso de não se usar divisão em janelas, a opção de janela deslizante não foi usada para a avaliação dos tempos de processamento, uma vez que os tempos de processamento são semelhantes com ou sem janela deslizante, pois a quantidade de informação musical é a mesma. A grande diferença é que não usando janela deslizante há um número de *reading frames* igual ao tamanho da palavra, e cada *reading frame* irá corresponder a uma pauta musical.

Divisão em janelas	Tamanho da janela	Janela deslizante	Tamanho da palavra	Conversão aminoácidos	Conversão *ECG*	**Tempo de processamento** (segundos)
Não	-	Não	1	Não	Não	1.1443
Não	-	Não	2	Não	Não	5.8260
Não	-	Não	3	Não	Não	8.5736
Não	-	Não	3	Sim	Não	12.3570
Não	-	Não	4	Não	Não	12.7396
Não	-	Não	4	Não	Sim	10.7411
Sim	60	-	1	Não	Não	1.8444
Sim	120	-	1	Não	Não	0.9314
Sim	240	-	1	Não	Não	0.5387
Sim	60	-	2	Não	Não	5.6210
Sim	60	-	3	Não	Não	5.8647
Sim	60	-	3	Sim	Não	5.9985
Sim	60	-	4	Não	Não	5.7338
Sim	60	-	4	Não	Sim	3.3586

Tabela 6.1: Tempos de processamento de conversão de uma sequência aleatória com 60000 nucleótidos em informação musical.

6.2 Distribuição das distâncias entre palavras e dos números de ordem

Serão apresentadas, como exemplo, duas distribuições de distâncias entre palavras e duas distribuições dos números de ordem das frequências de ocorrência das palavras:

- usando divisão em janelas;
- não usando divisão em janelas.

Também será apresentado para todos os casos o histograma de classes depois da função de otimização `optZipf`.

Foi usada a sequência de referência $NC\_014153.1$ (*Thiomonas intermedia K12 chromosome*) retirada de [NCBI, 2015]. No caso de não se usar a divisão em janelas usaram-se apenas os primeiros 60000 nucleótidos, apenas por uma questão de rapidez de execução não foram usados mais nucleótidos. No caso de se usar divisão em janelas foram usados apenas os primeiros 600000 nucleótidos (também apenas por uma questão de rapidez de execução).

Foi usado o tamanho da palavra 4 (sem conversão ECG) para ambos os casos, no caso de divisão em janelas o tamanho da janela usado foi 4000 nucleótidos (equivalente a 1000 tetra-nucleótidos). Considerou-se também que o número máximo de notas a usar e o número máximo de durações a usar é 8. Atribuiu-se os números de ordem das frequências de ocorrência às notas musicais e as distâncias entre palavras às durações. Portanto os números de ordem e as distâncias entre palavras serão agrupadas em 8 classes distintas $\{1, 2, ..., 8\}$ segundo a função `optZipf` (otimizando a lei de Zipf, em que as classes de menor valor correspondem à maior frequência de ocorrência).

Para ambos os casos (com e sem divisão em janelas) são mostrados gráficos de barras antes e depois da função `optZipf`, também é mostrado um diagrama de dispersão que avalia os dados após a realização da função `optZipf` com a reta de regressão e as estimativas do declive e do coeficiente de determinação.

A distribuição das distâncias no caso em que não se usou divisão em janelas é apresentada na figura 6.1, observa-se que as distâncias são mais frequentes em valores menores. No caso de se usar divisão em janelas, a distribuição de distâncias é mostrada na figura 6.2, é visível que a distância mais frequente corresponde a um valor próximo de 1000, que equivale ao tamanho da janela. À medida que o valor da distância se afasta de 1000 , a sua frequência de ocorrência vai diminuindo. Conclui-se então que a distribuição das distâncias, para o caso de divisão em janelas, é centrada em torno do valor do tamanho da janela. Tanto num caso como no outro, a função de otimização `optZipf` reagrupa as distâncias em classes de forma a que se ajustem bem à lei de Zipf.

A distribuição dos números de ordem, no caso sem divisão em janelas é apresentada na figura 6.3, no caso com divisão em janelas é apresentada na figura 6.4. Em ambos os casos, a função de otimização `optZipf` reagrupou os números de ordem de forma a que se adequem à lei de Zipf.

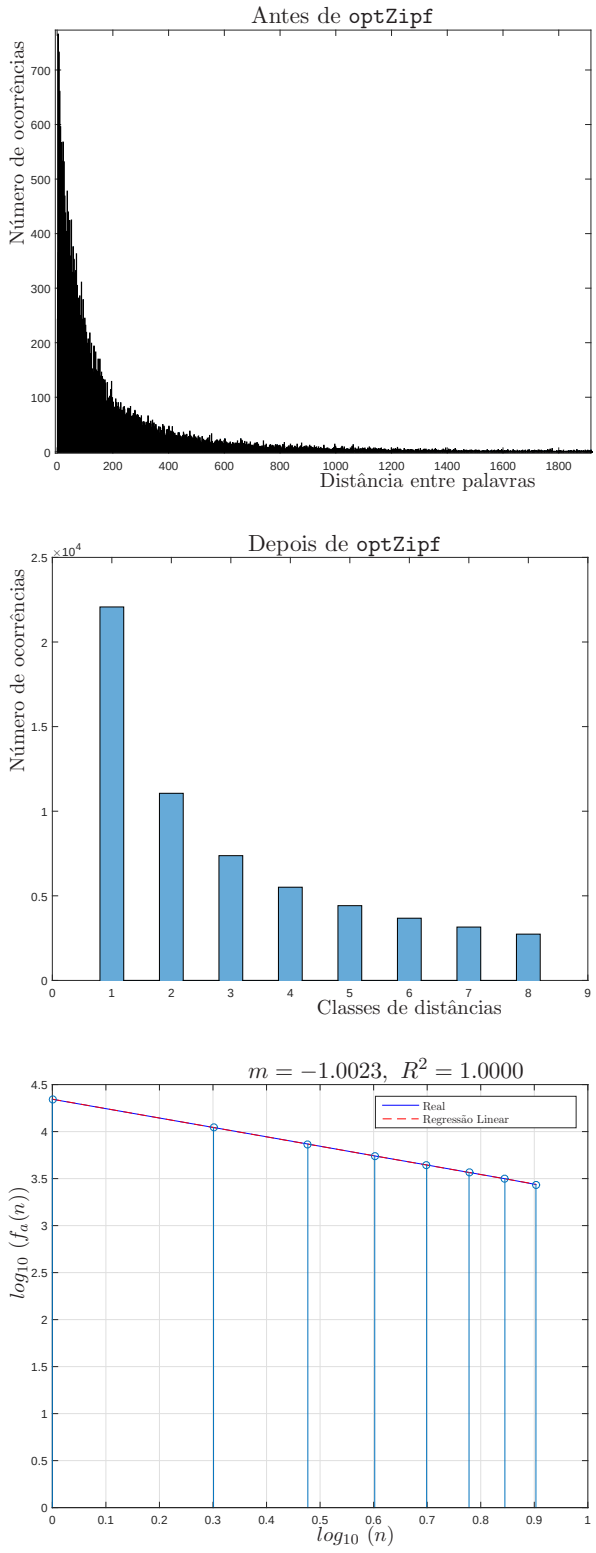

Figura 6.1: Distribuição das distâncias sem divisão em janelas com os primeiros 60000 nucleótidos da sequência de referência $NC\_014153.1$ [NCBI, 2015], usando tamanho da palavra 4.

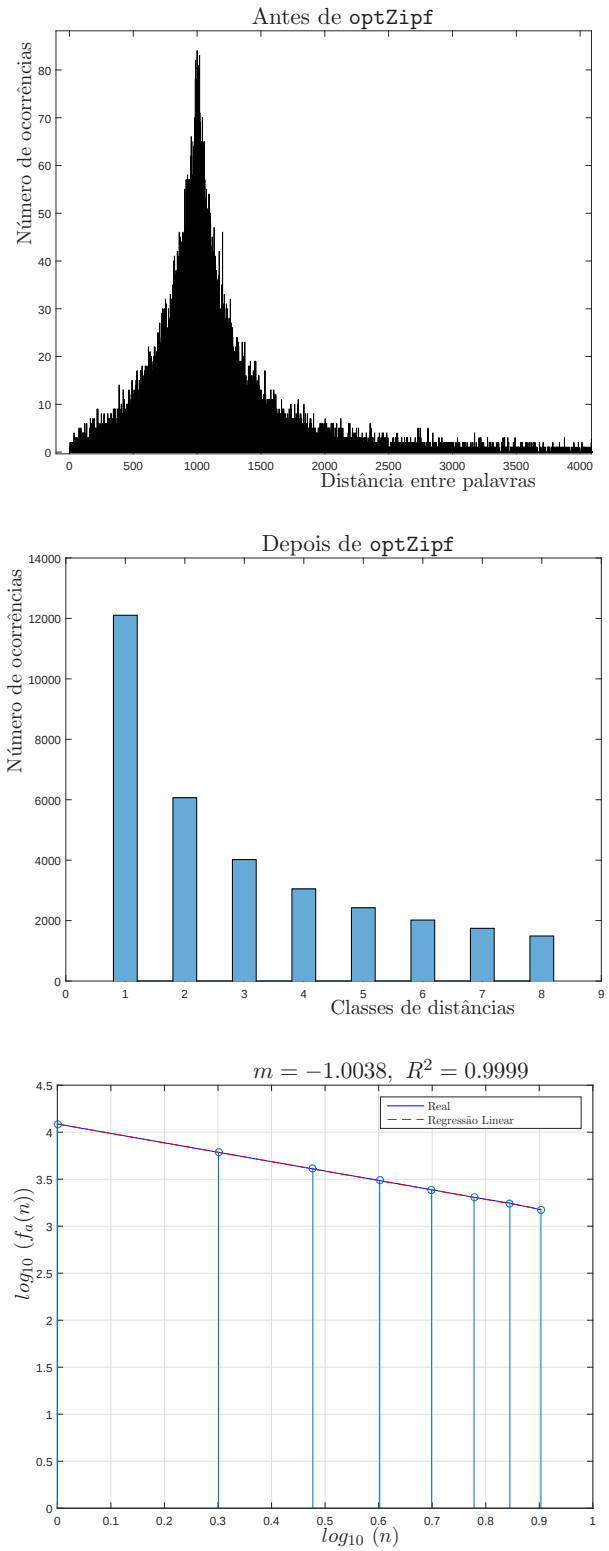

Figura 6.2: Distribuição das distâncias com divisão em janelas com os primeiros 600000 nucleótidos da sequência de referência $NC\_014153.1$ [NCBI, 2015], usando tamanho da palavra 4, e tamanho da janela 4000 nucleótidos.

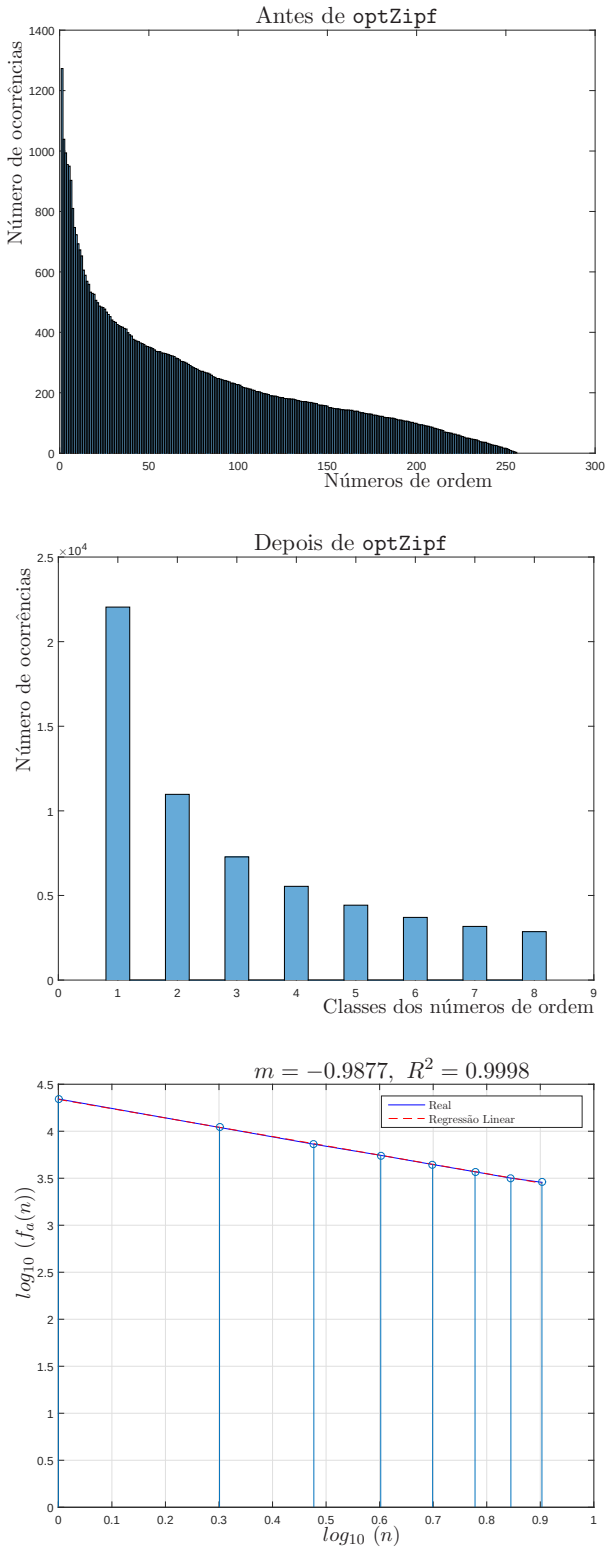

Figura 6.3: Distribuição dos números de ordem das frequências de ocorrência sem divisão em janelas com os primeiros 60000 nucleótidos da sequência de referência $NC\_014153.1$ [NCBI, 2015], usando tamanho da palavra 4.

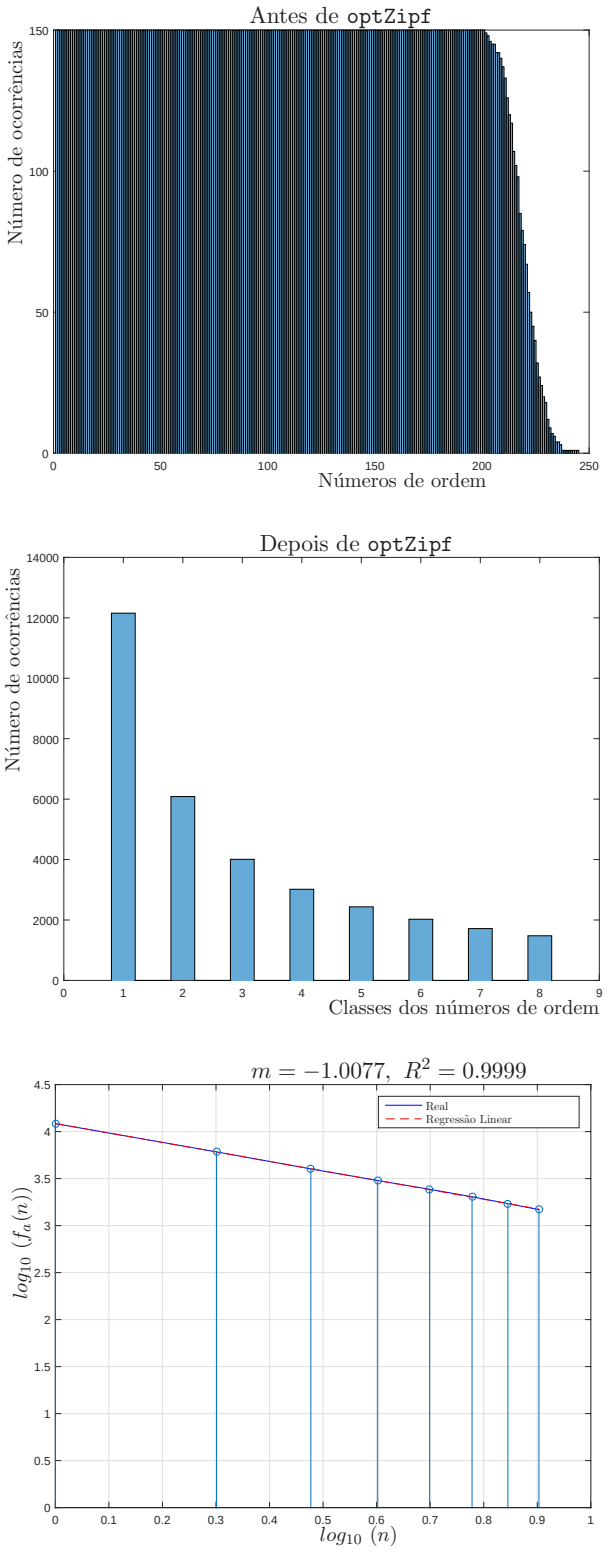

Figura 6.4: Distribuição dos números de ordem das frequências de ocorrência com divisão em janelas com os primeiros 600000 nucleótidos da sequência de referência $NC\_014153.1$ [NCBI, 2015], usando tamanho da palavra 4, e tamanho da janela 4000 nucleótidos.

6.3 Respostas a inquérito

Foi proposta a realização de um inquérito (ver questões do inquérito no anexo B) a cerca de uma centena de pessoas, das quais 48 realizaram o inquérito. Este inquérito teve como objetivo perceber se diferentes sequências simbólicas de ADN (obtidas de maneiras distintas), originam músicas totalmente diferentes. Também teve como objetivo perceber se há semelhanças entre a qualidade da música e as avaliações da lei de Zipf, assim como verificar se regiões código do ADN são propícias a produzir música mais agradável na perspetiva do ouvinte.

Para isso foram usadas quatro sequências distintas:

Música 1 (M1)
Sequência codificante (apenas regiões código) de um gene:
Dystrophin [Homo sapiens (human)] (EntrezGene ID: 1756 [Ensembl, 2015]*).*

Música 2 (M2)
Sequência não codificante (apenas regiões não-código) de um eucarionte:
Bos taurus mitochondrion breed Hanwoo [EMBL-EBI, 2015].

Música 3 (M3)
Sequência aleatória:
Sequência com 10000 nucleótidos com probabilidade de ocorrência equiprováveis.

Música 4 (M4)
Sequência completa (com regiões código e não-código) de um vírus:
Reston ebolavirus isolate RESTV/M.fascicularis-tc/PHL-USA/1996/Ferlite, Philippines/Alice, TX [EMBL-EBI, 2015].

Os parâmetros de conversão usados nas quatro sequências foram os mesmos influenciando de igual modo a criação da música:

- as notas musicais foram obtidas através dos números de ordem das frequências de ocorrência;
- as durações foram obtidas através das distâncias entre palavras;
- foi usado o tamanho da palavra 3 e conversão para aminoácidos, para fazer sentido nas regiões codificantes;
- não foi usada a divisão em janelas;
- número máximo de notas a usar: 21 (não usa a função de otimização de Zipf);
- número máximo de durações a usar: 8 (usa a função de otimização de Zipf);
- andamento 120, instrumento piano para as três pautas musicais (relativas às três *reading frames*).

Primeiramente foram feitos os seguintes estudos estatísticos (com recurso à lei de Zipf):

- avaliação do ajustamento da lei de Zipf nas notas musicais (individualmente);
- avaliação do ajustamento da lei de Zipf nas durações musicais (individualmente);
- avaliação do ajustamento da lei de Zipf nas notas musicais (com agrupamento de notas duas a duas, consiste no uso de uma janela deslizante de tamanho 2, esta avaliação foi também utilizada porque é um dos fatores musicais mais destacados do trabalho [Lo, 2012], foi concluído que usando janela deslizante de tamanho dois nas notas, a lei de Zipf é mais vezes respeitada nas músicas de grandes compositores).

Nas tabelas 6.2, 6.3 e 6.4 são mostrados respetivamente os resultados das propriedades da lei de Zipf (para cada uma das pautas musicais) para o caso das notas musicais, durações musicais e agrupamento de notas musicais duas a duas.

	Música 1			**Música 2**			**Música 3**			**Música 4**		
	m	R^2	$erro$	m	R^2	$erro$	m	R^2	$erro$	m	R^2	$erro$
1ª pauta	-0.92	0.47	0.54	-0.77	0.71	0.37	-0.61	0.83	0.42	-0.53	0.90	0.48
2ª pauta	-0.73	0.77	0.35	-0.77	0.70	0.38	-0.63	0.82	0.41	-0.52	0.83	0.51
3ª pauta	-0.76	0.63	0.44	-0.78	0.71	0.37	-0.62	0.82	0.42	-0.53	0.86	0.49

Tabela 6.2: Avaliação do ajustamento da lei de Zipf nas notas musicais (individualmente). m corresponde ao declive, R^2 corresponde ao coeficiente de determinação e o $erro$ corresponde à distância euclidiana relativamente ao ponto ideal $(-1, 1)$.

	Música 1			**Música 2**			**Música 3**			**Música 4**		
	m	R^2	$erro$	m	R^2	$erro$	m	R^2	$erro$	m	R^2	$erro$
1ª pauta	-1.11	0.97	0.11	-0.99	1.00	0.01	-0.96	0.99	0.04	-1.28	0.86	0.31
2ª pauta	-1.05	0.99	0.05	-1.00	1.00	0.00	-0.99	1.00	0.01	-1.03	1.00	0.03
3ª pauta	-1.13	0.96	0.14	-0.99	1.00	0.01	-0.95	1.00	0.05	-1.08	0.98	0.08

Tabela 6.3: Avaliação do ajustamento da lei de Zipf nas durações musicais (individualmente). m corresponde ao declive, R^2 corresponde ao coeficiente de determinação e o $erro$ corresponde à distância euclidiana relativamente ao ponto ideal $(-1, 1)$.

	Música 1			**Música 2**			**Música 3**			**Música 4**		
	m	R^2	$erro$	m	R^2	$erro$	m	R^2	$erro$	m	R^2	$erro$
1ª pauta	-0.90	0.67	0.34	-0.79	0.77	0.31	-0.77	0.75	0.34	-0.67	0.77	0.40
2ª pauta	-0.90	0.70	0.32	-0.81	0.77	0.30	-0.78	0.73	0.35	-0.63	0.77	0.44
3ª pauta	-0.87	0.65	0.37	-0.82	0.77	0.29	-0.78	0.73	0.35	-0.65	0.78	0.41

Tabela 6.4: Avaliação do ajustamento da lei de Zipf nas notas musicais (duas a duas). m corresponde ao declive, R^2 corresponde ao coeficiente de determinação e o $erro$ corresponde à distância euclidiana relativamente ao ponto ideal $(-1, 1)$.

As músicas 1 e 2 (referentes à região código do gene *Dystrophin*, e à região não-código de um eucarionte) são as que obtiveram declives mais próximos do ideal em todas as verificações de

Zipf. Já as músicas 3 e 4 (referentes a uma sequência aleatória e a uma sequência de um vírus) obtiveram coeficientes de determinação melhores. No geral as músicas 2 e 3 foram as que obtiveram menores erros (relativamente ao ponto ideal, ver secção 2.3 sobre o cálculo do erro através da distância euclidiana), portanto espera-se que estas músicas sejam ligeiramente mais agradáveis devido às suas propriedades gerais da lei de Zipf serem mais próximas do ideal relativamente às outras.

No formulário apenas foram apresentados os primeiros 30 segundos de cada música, sendo um dos objetivos o reconhecimento das músicas mais agradáveis. Para isso foi pedido aos inquiridos que avaliassem numa escala de 1 a 5 o quanto gostavam de cada música. Também foi pedido aos inquiridos que indicassem quais músicas consideravam aleatórias, com o objetivo de perceber se alguma sequência simbólica de ADN se sobressai com uma música aleatória (seria de esperar que a sequência aleatória fosse a que originasse a música "mais aleatória").

Foi testado se a agradabilidade das quatro músicas é igual, para isso foi usado o teste de *Friedman* e a hipótese de igualdade foi rejeitada (valor p = 0.009). Como o valor p tomou um valor inferior a 5% (nível de significância usado) conclui-se que a agradabilidade das várias músicas não é igual. A figura 6.5 apresenta quatro caixas de bigodes paralelas com os resultados da avaliação feita quanto à agradabilidade de cada música. As músicas 3 e 4 (respetivamente sequência aleatória e vírus) foram consideradas ligeiramente mais agradáveis que as músicas 1 e 2 (respetivamente sequência codificante e sequência não codificante). Curiosamente, os resultados não são completamente consistentes com o esperado, as músicas que gostaram mais foram as músicas 3 e 4, que correspondem a sequências com declives da lei de Zipf mais afastados do ideal.

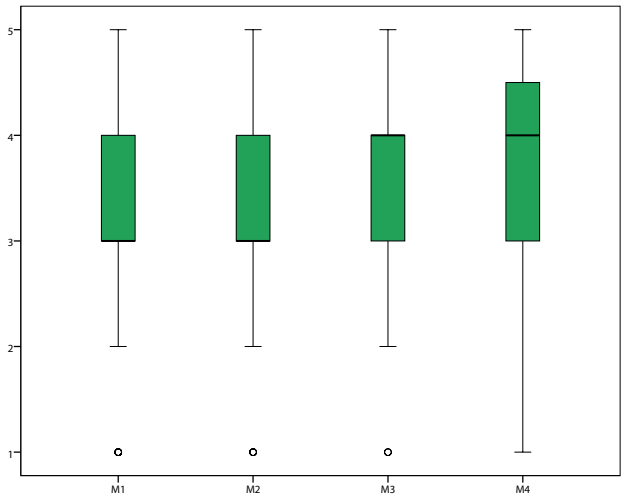

Figura 6.5: Resultados do inquérito: agradabilidade de cada música.

Foi obtido o número de inquiridos que classifica cada música como aleatória. Depois usou-se o teste de *Cochran* para testar se há um número semelhante de inquiridos a classificar cada música como aleatória. Não se rejeitou a hipótese de igualdade (valor p = 0.513), o que significa que do ponto de vista estatístico todas as músicas foram indicadas como aleatória

um número de vezes semelhante.

Na figura 6.6 apresenta-se a percentagem de vezes que cada música foi indicada como aleatória, mostrando que há uma pequena tendência para a música 3 ser mais frequentemente indicada como aleatória. No entanto, esta diferença pode não significar nada, e ser fruto do acaso, apenas com um número maior de respostas é que se obteriam resultados mais credíveis.

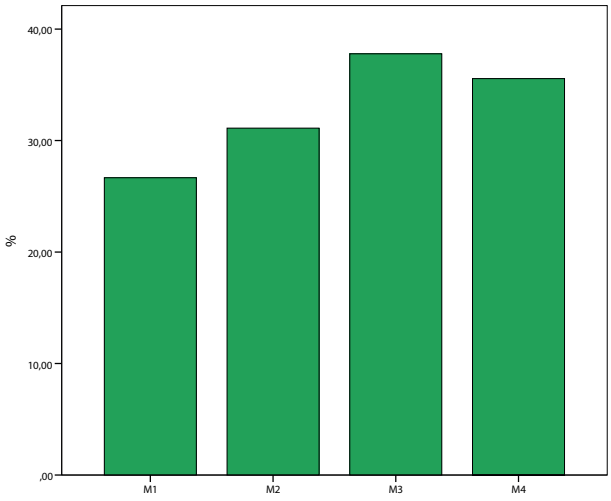

Figura 6.6: Resultados do inquérito: quais as músicas consideradas aleatórias.

Com as respostas deste inquérito torna-se evidente que há um longo trabalho a ser percorrido para que as músicas tenham mais sentido quando relacionadas com o ADN.

6.4 Análise comparativa entre regiões código e não-código

Esta secção tem como objetivo tentar diferenciar através da lei de Zipf, músicas criadas através de regiões código e músicas criadas através de regiões não-código. Para fazer este estudo usou-se apenas uma região código e uma região não-código em cada sequência de ADN usada, por minimizar o tempo de processamento. Para esta análise foram usadas cinco sequências de ADN (de [EMBL-EBI, 2015]), correspondentes a quatro bactérias e a um eucarionte:

- *bacillus subtilis subsp. subtilis str. 168* (Sequência 1 - S1);
- *chlamydia trachomatis D/UW-3/CX* (Sequência 2 - S2);
- *mycoplasma genitalium G37* (Sequência 3 - S3);
- *streptococcus mutans UA159* (Sequência 4 - S4);
- *homo sapiens mitochondrion isolate HeLa* (Sequência 5 - S5).

Foram usados os mesmos parâmetros de conversão nas cinco sequências de ADN para a criação da música, para não afetar na recolha de dados estatísticos. Os parâmetros foram:

- as notas musicais foram obtidas através dos números de ordem das frequências de ocorrência das palavras;
- as durações foram obtidas através das distâncias entre palavras;
- tamanho da palavra 2 (di-nucleótidos);
- não foi usada a divisão em janelas;
- foi usada janela deslizante;
- número máximo de notas a usar: 16 (não usa otimização segundo a lei de Zipf);
- número máximo de durações a usar: 8 (usa otimização segundo a lei de Zipf).

Foi avaliado o ajustamento da lei de Zipf para todas as sequências consideradas, usando apenas uma região código e uma região não-código. Foram avaliadas as seguintes caraterísticas:

- notas musicais (individualmente);
- durações musicais (individualmente).

Os resultados das propriedades da lei de Zipf para cada sequência, das notas musicais e durações musicais são mostrados respetivamente nas tabelas 6.5 e 6.6.

	S1		**S2**		**S3**		**S4**		**S5**	
	m	R^2	m	R^2	m	R^2	m	R^2	m	R^2
Região código	-0.47	0.94	-0.37	0.75	-1.11	0.83	-0.65	0.88	-0.71	0.77
Região não-código	-0.54	0.84	-0.52	0.76	-1.36	0.80	-1.01	0.65	-0.48	0.87

Tabela 6.5: Avaliação do ajustamento da lei de Zipf das notas musicais das músicas criadas a partir das 5 sequências de ADN mencionadas nesta secção. m corresponde ao declive e R^2 corresponde ao coeficiente de determinação.

	S1		**S2**		**S3**		**S4**		**S5**	
	m	R^2	m	R^2	m	R^2	m	R^2	m	R^2
Região código	-0.95	0.99	-0.89	0.99	-1.31	0.84	-0.99	1.00	-1.18	0.92
Região não-código	-1.26	0.85	-1.05	0.98	-1.37	0.81	-0.92	0.98	-1.07	0.98

Tabela 6.6: Avaliação do ajustamento da lei de Zipf das durações musicais das músicas criadas a partir das 5 sequências de ADN mencionadas nesta secção. m corresponde ao declive e R^2 corresponde ao coeficiente de determinação.

Através da visualização das tabelas, pode-se concluir que uma determinada sequência numa determinada caraterística (notas ou durações) possui valores de declive e coeficiente de determinação semelhantes entre a região código e a região não-código. De um modo geral não

existe diferenças significativas entre o ajustamento da lei de Zipf numa região código ou numa região não-código. As durações têm uma distribuição de probabilidades mais próxima à lei de Zipf quando comparadas com as notas, isto porque a função da otimização de Zipf é usada nas durações e não é usada nas notas.

Neste trabalho não se conseguiu tirar informação relevante de uma música para caraterizar alguma área do ADN, pelo que é necessário realizar outros mapeamentos ou usar outros processos para conseguir reter informação relevante de uma música.

Capítulo 7

Conclusões e trabalho futuro

Nesta dissertação foram apresentados mapeamentos entre símbolos, números e música, tendo fundamentalmente por base as distâncias entre palavras e as frequências de ocorrência das palavras.

Foram criados essencialmente dois algoritmos distintos de conversão de ADN em música:

- sem divisão em janelas: não faz divisão em janelas (se usar janela deslizante é criada uma única pauta musical, caso contrário o número de pautas musicais criadas corresponde ao número de *reading frames*, que é igual ao tamanho da palavra usado);
- com divisão em janelas: faz divisão em janelas, a cada palavra distinta corresponde uma pauta musical única.

O algoritmo sem divisão em janelas permite "ouvir simultaneamente várias *reading frames*" (até 4, no caso de tamanho da palavra 4), enquanto que o algoritmo com divisão em janelas permite "ouvir simultaneamente várias palavras" (ou seja, ouve-se a "música genómica" associada a cada palavra).

Estes algoritmos possuem vários parâmetros que podem ser alterados (tamanho da palavra, quantas e quais notas/durações usar, andamento da música, etc), permitindo criar várias músicas distintas a partir de uma sequência simbólica, em particular a partir de uma sequência de ADN. Nalgumas combinações dos parâmetros de entrada o algoritmo depende muito da função de otimização de Zipf, devido a isto as probabilidades de ocorrência das notas e durações irão tender a ter uma distribuição de probabilidades segundo a lei de Zipf. Este procedimento não é totalmente positivo, uma vez que qualquer sequência simbólica de ADN gera informação com probabilidades semelhantes, originando assim músicas semelhantes. Ou seja, a unicidade e exclusividade de cada sequência acaba por ser distorcida.

Uma sugestão de alteração para trabalho futuro seria aplicar diferentes processos de conversão, podendo ser usado um determinado avaliador musical objetivo, por forma a criar música mais agradável da perspetiva do ouvinte.

Outra ideia passa por desenvolver novos algoritmos mais robustos e coerentes por forma a conseguir tirar resultados objetivos sobre determinadas caraterísticas do ADN, o que não foi conseguido nesta dissertação. Deseja-se conseguir a deteção de padrões, a diferenciação entre

espécies através da música, a distinção entre regiões código e não-código e a capacidade de perceber se se trata de uma sequência aleatória.

Bibliografia

[Afreixo et al., 2009] Afreixo, V., Bastos, C. A., Pinho, A. J., Garcia, S. P., and Ferreira, P. J. (2009). Genome analysis with inter-nucleotide distances. *Bioinformatics*, 25(23):3064–3070.

[Afreixo et al., 2014] Afreixo, V., Rodrigues, J. M., and Bastos, C. A. (2014). Exceptional single strand dna word symmetry: Universal law? In *8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014)*, pages 137–143. Springer.

[EMBL-EBI, 2015] EMBL-EBI (consultado em Maio de 2015). *The European Bioinformatics Institute*. `http://www.ebi.ac.uk/genomes/`.

[Ensembl, 2015] Ensembl (consultado em Maio de 2015). `http://www.ensembl.org/`.

[Gena and Strom, 1995] Gena, P. and Strom, C. (1995). Musical synthesis of dna sequences. *XI Colloquio di Informatica Musicale*, pages 203–204. Bologna: Universita di Bologna.

[Henrique, 2002] Henrique, L. L. (2002). *Acústica musical*. Fundação Calouste Gulbenkian, Lisboa.

[Hill, 1970] Hill, B. M. (1970). Zipf's law and prior distributions for the composition of a population. *Journal of the American Statistical Association*, 65(331):1220–1232.

[Ingallsa et al., 2009] Ingallsa, T., Martiusb, G., Hellmuthc, M., Marzc, M., and Prohaskac, S. J. (2009). Converting dna to music: Composalign. page 93.

[Lo, 2012] Lo, M. Y. (2012). *Evolving cellular automata for music composition with trainable fitness functions*. PhD thesis, University of Essex.

[Manaris et al., 2005] Manaris, B., Romero, J., Machado, P., Krehbiel, D., Hirzel, T., Pharr, W., and Davis, R. B. (2005). Zipf's law, music classification, and aesthetics. *Computer Music Journal*, 29(1):55–69. Citado por [Lo, 2012].

[Manaris et al., 2003] Manaris, B., Vaughan, D., Wagner, C., Romero, J., and Davis, R. B. (2003). Evolutionary music and the zipf-mandelbrot law: Developing fitness functions for pleasant music. In *Applications of Evolutionary Computing*, pages 522–534. Springer. Citado por [Lo, 2012].

[Miranda, 2002] Miranda, E. (2002). *Computer sound design: synthesis techniques and programming*. Taylor & Francis. Oxford, UK: Focal Press.

[MMA, 2015] MMA (consultado em Maio de 2015). *Midi Manufacturers Association*. `http://www.midi.org/`.

[Nair and Mahalakshmi, 2005] Nair, A. S. S. and Mahalakshmi, T. (2005). Visualization of genomic data using inter-nucleotide distance signals. *Proceedings of IEEE Genomic Signal Processing*, 408. Bucharest, Romania.

[NCBI, 2015] NCBI (consultado em Maio de 2015). *National Center for Biotechnology Information*. `http://www.ncbi.nlm.nih.gov/nuccore/`.

[Ohno, 1987] Ohno, S. (1987). Repetition as the essence of life on this earth: music and genes. In *Haematology and Blood Transfusion*, pages 511–518. Springer.

[Ohno, 1993] Ohno, S. (1993). A song in praise of peptide palindromes. *Leukemia*, 7:S157–9.

[Ohno and Ohno, 1986] Ohno, S. and Ohno, M. (1986). The all pervasive principle of repetitious recurrence governs not only coding sequence construction but also human endeavor in musical composition. *Immunogenetics*, 24(2):71–78.

[Petersen, 2001] Petersen, M. (July 2001). Mathematical harmonies.

[Petersen, 2004] Petersen, M. R. (2004). Musical analysis and synthesis in matlab. *College Mathematics Journal*, pages 396–401.

[Russ, 2004] Russ, M. (2004). *Sound synthesis and sampling*. Taylor & Francis. Focal Press.

[Sánchez Sousa et al., 2005] Sánchez Sousa, A., Baquero, F., and Nombela, C. (2005). The making of the genoma music. *Revista iberoamericana de micología*, 22(4):242–248.

[Schutte, 2015] Schutte, K. (2012, consultado em Janeiro de 2015). *MATLAB and MIDI*. `http://www.kenschutte.com/midi`.

[Shtrikman, 1994] Shtrikman, S. (1994). Some comments on zipf's law for the chinese language. *Journal of Information Science*, 20(2):142–143. Citado por [Lo, 2012].

[Takahashi and Miller, 2007] Takahashi, R. and Miller, J. H. (2007). Conversion of amino-acid sequence in proteins to classical music: search for auditory patterns. *Genome biology*, 8(5):405.

[Watson and Crick, 1953] Watson, J. D. and Crick, F. H. (1953). Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738.

[Zanette, 2006] Zanette, D. H. (2006). Zipf's law and the creation of musical context. *Musicae Scientiae*, 10(1):3–18. Citado por [Lo, 2012].

[Zipf, 1935] Zipf, G. K. (1935). The psycho-biology of language. Houghton-Mifflin. Citado por [Lo, 2012].

[Zipf, 1949] Zipf, G. K. (1949). Human behavior and the principle of least effort. Addison-Wesley. Citado por [Lo, 2012].

Apêndice A

Código MATLAB

Neste anexo apresentam-se todas as funções desenvolvidas em MATLAB e uma breve explicação do funcionamento de cada uma. Para um melhor entendimento das funções é necessária a compreensão de todo o código desenvolvido em MATLAB.

- `adjustDistances`

  Parâmetros de entrada:

  - Sequência de nucleótidos;
  - Número da palavra;
  - Tamanho da janela;
  - Tamanho da palavra.

  Parâmetros de saída:

  - Distâncias respetivas da palavra usada;
  - Posições da palavra.

  Operação: Determina as distâncias e respetivas posições de uma determinada palavra usando divisão em janelas.

- `calcDistanceZipf`

  Parâmetros de entrada:

  - Declive (m);
  - Coeficiente de determinação (R^2).

  Parâmetros de saída:

  - Erro (distância euclidiana).

  Operação: Determina a distância euclidiana do ponto $\left(m, R^2\right)$ ao ponto ideal $(-1, 1)$.

- `calcDists`

  Parâmetros de entrada:

  - Sequência de dados;
  - Vetor contendo informação sobre o tipo de conversão, tamanho da palavra (indica se usa conversão para aminoácidos ou *ECG* ou se não faz qualquer conversão) e tamanho da janela.

  Parâmetros de saída:

  - Distâncias;
  - Posições da palavra (caso se use divisão em janelas).

  Operação: Caso use divisão em janelas, invoca a função `adjustDistances` e determina as distâncias e respetivas posições de uma determinada palavra, caso contrário, invoca a função `newCalcDistances` e determina apenas as distâncias.

- `calcErrors`

  Parâmetros de entrada:

  - Sequência de dados;
  - Número da palavra;
  - Tamanho da janela;
  - Posições da palavra na sequência;
  - Tamanho da palavra.

  Parâmetros de saída:

  - Vetor de erros.

  Operação: É usada no caso de divisão em janelas, e calcula os erros respetivos a uma determinada palavra, que estão relacionados com o grau de afastamento entre a frequência local da palavra (numa determinada janela) e a frequência global da palavra. Estes erros são usados para a determinação das intensidades musicais.

- `calcFOC`

  Parâmetros de entrada:

  - Sequência de dados;
  - Número da palavra;
  - Tamanho da janela;
  - Posições da palavra na sequência;
  - Tamanho da palavra.

  Parâmetros de saída:

  - Vetor de números de ordem das frequências de ocorrência das palavras.

Operação: É usado no caso de divisão em janelas, atribui a cada palavra da sequência um número associado com a sua frequência de ocorrência (em que o número 1 significa que é o mais frequente, o número 2 que é o segundo mais frequente, etc.).

- `calcIntensities`

  Parâmetros de entrada:

  - Sequência de dados;
  - Tamanho da palavra;
  - Indicação se usa conversão para aminoácidos ou *ECG* ou se não faz qualquer conversão.

  Parâmetros de saída:

  - Vetor de intensidades.

  Operação: É usado no caso em que não há divisão em janelas, calcula as intensidades através das probabilidades de ocorrência de cada palavra numa determinada janela.

- `calcNonCodeRegions`

  Parâmetros de entrada:

  - Matriz de números que indicam onde iniciam e terminam as regiões código;
  - Tamanho da sequência (número de nucleótidos).

  Parâmetros de saída:

  - Matriz de números que indicam onde iniciam e terminam as regiões não-código;
  - Matriz de números que indicam onde iniciam e terminam as regiões código e não-código.

  Operação: Através das posições de início e fim das regiões código determinam-se as posições de início e fim das regiões não-código.

- `calcSeqs`

  Parâmetros de entrada:

  - Sequência de dados (apenas nucleótidos);
  - Vetor contendo informação sobre o tipo de conversão, tamanho da palavra (indica se usa conversão para aminoácidos ou *ECG* ou se não faz qualquer conversão) e indicação se usa janela deslizante.

  Parâmetros de saída:

  - Sequência de dados com os números corretos (consoante o tamanho da palavra, etc.).

  Operação: Converte a sequência de nucleótidos numa sequência equivalente (respeitando o tamanho da palavra, janela deslizante, etc.).

- `calcStartEndTime`

  Parâmetros de entrada:

  - Durações;
  - Tempo inicial de pausa;
  - Andamento.

  Parâmetros de saída:

  - Tempos de início;
  - Tempos de fim.

  Operação: Através das durações de cada nota musical calcula os respetivos tempo de início e tempo de fim de cada nota.

- `convertAsciiToCorrect`

  Parâmetros de entrada:

  - Sequência de nucleótidos (com numeração ASCII).

  Parâmetros de saída:

  - Sequência de nucleótidos (números 1 a 4).

  Operação: Converte a sequência com numeração ASCII para uma sequência de números 1 a 4.

- `convertDataToInfo`

  Parâmetros de entrada:

  - Sequência de dados;
  - Vetor contendo informação sobre o tipo de conversão, tamanho da palavra (indica se usa conversão para aminoácidos ou *ECG* ou se não faz qualquer conversão), tamanho da janela e indicação se usa janela deslizante;
  - Vetor de notas a usar;
  - Vetor de durações a usar;
  - Número de notas a usar;
  - Número de durações a usar.

  Parâmetros de saída:

  - Informação musical (estrutura com as notas, durações e intensidades);
  - Sequência de dados;
  - Distâncias;
  - Vetor de números de ordem das frequências de ocorrência das palavras.

  Operação: Converte a sequência de dados (segundo vários parâmetros) numa estrutura que contém informação musical.

- `convertMusicalInfoToMidi`

  Parâmetros de entrada:

  - Informação musical;
  - Nome do ficheiro a gravar;
  - Andamento;
  - Instrumentos a usar;
  - Indicação das pautas a ouvir;
  - Indicação das regiões a ouvir;
  - Indicação se é usado marcador sonoro entre regiões.

  Parâmetros de saída:

  - Matriz que contém parâmetros relativos à informação MIDI: as notas, intensidades, etc;
  - Indicação de quantos segundos demora cada região.

  Operação: Cria um ficheiro no formato MIDI (.mid) através da informação musical e restantes parâmetros de entrada.

- `convertRegionsToMusicalInfo`

  Parâmetros de entrada:

  - Indicação se faz distinção entre regiões (se sim, quais usar);
  - Sequência de dados;
  - Posições de início e fim de regiões código.
  - Vetor contendo informação sobre o tipo de conversão, tamanho da palavra (indica se usa conversão para aminoácidos ou *ECG* ou se não faz qualquer conversão), tamanho da janela e indicação se usa janela deslizante;
  - Vetor de notas a usar;
  - Vetor de durações a usar;
  - Número de notas a usar;
  - Número de durações a usar.

  Parâmetros de saída:

  - Informação musical (estrutura que contém as notas, durações, intensidades e indicação de que tipo de região se trata, código, não-código, ou ainda se não faz qualquer distinção de regiões);
  - Número de regiões ignoradas devido a um número de nucleótidos insuficiente.

  Operação: Faz uso da função `convertDataToInfo`, converte uma sequência de dados (segundo vários parâmetros) numa estrutura que contém informação musical (esta estrutura pode conter várias regiões[1].

[1] Neste sentido entenda-se por região a música respetiva de uma determinada região de um gene.

- `convertThreeNucleotidToAminoacid`

  Parâmetros de entrada:

  - Números de três nucleótidos.

  Parâmetros de saída:

  - Número do aminoácido respetivo.

  Operação: Converte três números consecutivos (relativos a três nucleótidos) num único número que codifica um aminoácido.

- `convertToX`

  Parâmetros de entrada:

  - Sequência de dados;
  - Tamanho da palavra;
  - Indicação se usa conversão para aminoácidos ou *ECG* ou se não faz qualquer conversão.

  Parâmetros de saída:

  - Sequência de dados (com numeração correta, respeitando as conversões usadas).

  Operação: Converte uma sequência de números (que indicam nucleótidos), noutra sequência de números que indicam as palavras respetivas (por exemplo di-nucleótidos, aminoácidos, etc).

- `detectTypeFile`

  Parâmetros de entrada:

  - Nome do ficheiro.

  Parâmetros de saída:

  - Tipo de ficheiro.

  Operação: Indicação se o ficheiro é do tipo FASTA, do tipo TEXT do site [EMBL-EBI, 2015] ou se não é nenhum dos tipos anteriores.

- `f_ecg`

  Parâmetros de entrada:

  - Vetor de nucleótidos a converter.

  Parâmetros de saída:

  - Número do símbolo *ECG* respetivo.

  Operação: Converte a numeração dos nucleótidos em numeração *ECG*.

- `f2`

  Parâmetros de entrada:

  - Dois números (nucleótidos).

  Parâmetros de saída:

  - Índice do di-nucleótido.

  Operação: Converte dois nucleótidos num di-nucleótido.

- `f3`

  Parâmetros de entrada:

  - Três números (nucleótidos).

  Parâmetros de saída:

  - Índice do tri-nucleótido.

  Operação: Converte três nucleótidos num tri-nucleótido.

- `f4`

  Parâmetros de entrada:

  - Quatro números (nucleótidos).

  Parâmetros de saída:

  - Índice do tetra-nucleótido.

  Operação: Converte quatro nucleótidos num tetra-nucleótido.

- `matrix2midi` (Adaptado do trabalho [Schutte, 2015])

  Parâmetros de entrada:

  - Matriz contendo informação musical: notas, intensidades, canais, etc.;
  - Vetor de instrumentos a usar.

  Parâmetros de saída:

  - Estrutura de formato MIDI.

  Operação: Gera uma estrutura MIDI através de uma matriz especificando toda a informação musical. Essa estrutura é usada por a função `writemidi` para gerar um ficheiro do tipo .mid.

- `new_gui_page0`

  Operação: Realiza a primeira página da interface, permite a escolha de três opções:

  - Conversão de dados em informação musical;
  - Criação da música (MIDI) através da informação musical;
  - Avaliação do ajustamento da lei de Zipf.

- `new_gui_page1`

  Operação: Realiza a segunda página da interface, possibilita a conversão de dados simbólicos segundo vários parâmetros em informação musical (gravada num ficheiro do tipo .txt).

- `new_gui_page2`

  Operação: Realiza a terceira página da interface, possibilita a criação de música (num ficheiro do tipo .mid) segundo vários parâmetros através da informação musical (gravada num ficheiro do tipo .txt).

- `new_gui_page3`

  Operação: Realiza a quarta página da interface, permite avaliar o ajustamento da lei de Zipf de uma determinada música (gravada num ficheiro do tipo .txt).

- `newCalcDistances`

  Parâmetros de entrada:

  - Sequência de dados.

  Parâmetros de saída:

  - Distâncias.

  Operação: É usada no caso em que não se usa divisão em janelas, através da sequência de dados determina as distâncias respetivas.

- `optZipf`

  Parâmetros de entrada:

  - Vetor de dados;
  - Valor que indica quantos são os números distintos desejados.

  Parâmetros de saída:

  - Novo vetor de dados otimizado.

  Operação: Reagrupa o vetor de dados obedecendo à distribuição da lei de Zipf, até ter um determinado número distinto de valores. Agrupa vários valores em classes de forma a que as classes de menor valor estejam associadas a maiores probabilidades.

- `preZipf`

  Parâmetros de entrada:

  - Vetor de dados;
  - Indicação se ignora a ordenação ou não.

  Parâmetros de saída:

  - Contagem dos dados.

  Operação: Conta quantas vezes aparece cada símbolo e devolve o vetor de contagens.

- `pte_convertDataToInfo`

  Parâmetros de entrada:

  - Número de nucleótidos da sequência;
  - Vetor contendo informação sobre o tipo de conversão, tamanho da palavra (indica se usa conversão para aminoácidos ou *ECG* ou se não faz qualquer conversão), tamanho da janela e indicação se usa janela deslizante.

  Parâmetros de saída:

  - Tempo de processamento máximo.

  Operação: Dado o tamanho da sequência e os parâmetros de conversão faz uma estimação grosseira do tempo máximo de processamento da função `convertDataToInfo`.

- `pte_convertMusicalInfoToMidi`

  Parâmetros de entrada:

  - Estrutura que contém a informação musical;
  - Indicação das pautas a ouvir;
  - Indicação das regiões a ouvir.

  Parâmetros de saída:

  - Tempo de processamento máximo.

  Operação: Dada a informação musical, as pautas e regiões a ouvir faz uma estimação razoável do tempo máximo de processamento da função `convertMusicalInfoToMidi`.

- `pte_convertRegionsToMusicalInfo`

  Parâmetros de entrada:

  - Indicação se faz distinção entre regiões ou não (se sim, quais regiões usa);
  - Sequência de dados;
  - Valores que indicam o início e fim de cada região código;
  - Vetor contendo informação sobre o tipo de conversão, tamanho da palavra (indica se usa conversão para aminoácidos ou *ECG* ou se não faz qualquer conversão), tamanho da janela e indicação se usa janela deslizante.

Parâmetros de saída:

- Tempo de processamento máximo.

Operação: Dada a indicação de que regiões usa (no caso de usar regiões), ou a indicação de que usa a sequência toda, calcula o número total de nucleótidos usados, e então usa a função `pte_convertDataToInfo` para estimar o tempo máximo de processamento da função `convertRegionsToMusicalInfo`.

- `read_file` (Adaptado de uma versão desenvolvida pelos orientadores)

  Parâmetros de entrada:

  - Nome do ficheiro.

  Parâmetros de saída:

  - Sequência de dados;
  - Matriz contendo valores que indicam o início e fim das regiões código.

  Operação: Lê ficheiros do tipo TEXT do site [EMBL-EBI, 2015].

- `readFile`

  Parâmetros de entrada:

  - Nome do ficheiro.

  Parâmetros de saída:

  - Sequência de dados.

  Operação: Lê ficheiros do tipo FASTA (não faz distinção entre regiões, lê apenas a sequência).

- `readmidi` (Adaptado do trabalho [Schutte, 2015])

  Parâmetros de entrada:

  - Nome do ficheiro (do tipo .mid).

  Parâmetros de saída:

  - Estrutura contendo informação acerca do ficheiro gravado em MIDI.

  Operação: Lê a informação de um ficheiro do tipo .mid e armazena numa estrutura.

- `readMusicalInfo`

  Parâmetros de entrada:

  - Nome do ficheiro.

  Parâmetros de saída:

  - Informação musical.

Operação: Lê a informação musical de um ficheiro (do tipo .txt) e armazena numa estrutura.

- `returnSoundMatrix`

  Parâmetros de entrada:

  - Tipo da região (código, não-código, ou não faz distinção).

  Parâmetros de saída:

  - Matriz que contém a informação musical do marcador sonoro respetivo;
  - Duração do som.

  Operação: Devolve a matriz que contém a informação musical (para usar na função `matrix2midi`) de um determinado som associado a um tipo de região.

- `verifyZipf`

  Parâmetros de entrada:

  - Declive (m);
  - Coeficiente de determinação (R^2).

  Parâmetros de saída:

  - Indicação se verifica ou não a lei de Zipf.

  Operação: Valida-se a lei de Zipf se $-1.2 \leq m \leq -0.8$ e $R^2 \geq 0.7$.

- `writemidi` (Adaptado do trabalho [Schutte, 2015])

  Parâmetros de entrada:

  - Estrutura contendo informação MIDI;
  - Nome do ficheiro.

  Operação: Escreve num ficheiro (do tipo .mid) a informação MIDI que está armazenada numa estrutura.

- `writeMusicalInfo`

  Parâmetros de entrada:

  - Estrutura que contém a informação musical;
  - Nome do ficheiro.

  Operação: Escreve o conteúdo da estrutura que contém a informação musical num ficheiro (do tipo .txt).

- `zipf`

  Parâmetros de entrada:

  - Vetor de contagens.

  Parâmetros de saída:

  - Declive (m);
  - Coeficiente de determinação (R^2).

  Operação: É aplicado o gráfico log-log ao vetor de contagens e utilizado o método dos mínimos quadrados para aproximar os dados a uma reta de primeiro grau, obtendo assim o declive e o coeficiente de determinação.

Apêndice B

Inquérito

Neste anexo são apresentadas as perguntas que foram realizadas no formulário sobre quatro músicas distintas, que foram convertidas a partir de sequências diferentes de ADN.

As perguntas realizadas foram:

1. O quanto gostou da música 1? (1 - detestar, 5 - adorar)
2. O quanto gostou da música 2?
3. O quanto gostou da música 3?
4. O quanto gostou da música 4?
5. Alguma das músicas lhe pareceu aleatória? Se sim, selecione a(s) música(s) que lhe pareceram aleatórias:
    - Música 1
    - Música 2
    - Música 3
    - Música 4
    - Nenhuma